Universidad de Cantabria

FACULTAD DE CIENCIAS

Support Vector Machines for Binary Classification: Theory and Practice

(Máquinas de vector soporte para clasificación binaria: teoría y práctica)

Trabajo de fin de Grado
para acceder al

GRADO EN MATEMÁTICAS

Autor: Sergio Remírez Miquélez

Directora: Cecilia Pola Méndez

Julio - 2023

For those who have been my aid and inspiration throughout this rewarding process. This would not have been possible without you. And to those that are no longer here, your memory lives on in my heart.

(A quienes han sido mi apoyo e inspiración a lo largo de este tiempo. Esto no hubiese sido posible sin vosotros. Y para aquellos que ya no están, vuestro recuerdo perdura en mí.)

Abstract

In this paper we study support vector machines (SVM) for binary classification. We will start with the case where the data set of the classification problem is linearly separable, for which we will consider hard margin SVM. We will continue with the case where the data set is not necessarily linearly separable, for which we will use soft margin SVM. In both cases we will formulate a quadratic optimization problem and we will obtain an alternative formulation using Wolfe duality.

The final theoretical part of this paper consists in using kernel functions to improve the versatility of support vector machines for the case where the data set is not linearly separable.

Finally, we have done an experimental part applying the Scikit-learn library in Python to "Breast Cancer Wisconsin (Diagnostic) Data Set". Moreover, Matlab has been used for some illustrative examples.

Key words: support vector machine (SVM), binary classification, hard margin SVM, soft margin SVM, Wolfe duality, Kernel function.

Resumen

En este trabajo estudiamos las máquinas de vector soporte (SVM por sus siglas en inglés) para clasificación binaria. Comenzaremos con el caso en el que el conjunto de datos del problema de clasificación es linealmente separable, para el cuál consideraremos las SVM con margen duro. Continuaremos con el caso en el que el conjunto de datos no es necesariamente linealmente separable, para el que usaremos las SVM con margen blando. En ambos casos formularemos un problema de optimización cuadrática y obtendremos una formulación alternativa mediante la dualidad de Wolfe.

La última parte teórica de este trabajo consiste en utilizar las funciones kernel para potenciar la versatilidad de las máquinas de vector soporte para el caso en que los datos no sean separables linealmente.

Por último, hemos realizado una parte experimental utilizando la librería Scikit-learn de Python y la base de datos "Breast Cancer Wisconsin (Diagnostic) Data Set". Además, Matlab ha sido usado para algunos ejemplos ilustrativos.

Palabras clave: máquina de vector soporte, clasificación binaria, SVM con margen duro, SVM con margen blando, dualidad de Wolfe, función kernel.

Contents

Chapter 1

Introduction

Support Vector Machines (SVM) achieved greater popularity and recognition in the 1990s when V. Vapnik and his co-workers used kernel functions to enhance the versatility of SVM, see [5] and references therein. This supervised machine learning method has been used in several fields such as text categorization, image recognition, digit recognition, or non-stationary signal classification.

Although SVM were originally created to solve binary classification problems, they were reformulated to solve other kind of problems, not discussed in this paper, such us regression (see Chapter 9 of [12]) or multiclassification (see Section 7.6 of [12]).

Mathematical perspective

Before explaining and formalizing the general idea, let us consider an application of SVM to illustrate the objective of this paper. Let us suppose that we have a set of m tumors and for each of them we have measured n certain characteristics and we know whether they are benign or malignant. The problem we pose is: Can we find a function that helps us to predict if a new tumor is benign or malignant by measuring the same characteristics as the ones we already have? This is what we will discuss throughout this paper. That is, we will try to find a function that helps us to classify new tumors.

Now, let us consider a general case. Let us suppose that we have a finite set of data points $\{x_1, x_2, ..., x_m\} \subset \mathbb{R}^n$, which can be divided into two classes depending on whether they have a certain property or not. It is clear that we can represent the data set as follows

$$S = \{(x_i, y_i) : x_i \in \mathbb{R}^n, y_i \in \{-1, 1\}, i = 1, ..., m\}.$$

We will named it as **binary data set** and x_i as **data points**. If we use the previous

notation, we can also consider that the **classes** are

$$S_1 = \{x_i : (x_i, 1) \in S\} \quad \text{and} \quad S_{-1} = \{x_i : (x_i, -1) \in S\}.$$

We obviously assume that $S_1 \neq \emptyset$, $S_{-1} \neq \emptyset$ and $S_1 \cap S_{-1} = \emptyset$.

The first approach to the problem consists in assuming that the binary data set is linearly separable. So, we can find a hyperplane that completely separates the classes S_1 and S_{-1} and, then we can define a classification function. In the next step we will adapt the results obtained for the previous case to general binary data sets.

Structure of the paper

Now, we will briefly summarize each chapter.

- **Chapter 2**. We will consider the case in which the classes S_1 and S_{-1} are linearly separable by a hyperplane. Of the innumerable hyperplanes, we will consider which of them best separates the classes S_1 and S_{-1}, in some way. For this purpose we will construct a convex quadratic optimization problem that is known as hard margin SVM problem. We will denote it as (P_{HM}). We will prove the existence and uniqueness of a global solution for (P_{HM}). Then, we will define a classification function, which as its name suggests, it will help us to classify new points.

  Then, we will use Wolfe duality to obtain an alternative formulation of (P_{HM}), which we will denote as (D_{HM}). The objective is to find a problem (D_{HM}) that will be less expensive to solve than (P_{HM}). We will prove the non-uniqueness of solution.

  We will also define support vectors, which give name to the SVM method, and they are those that mainly define the classification function ("machine").

- **Chapter 3**. In this chapter, we will assume that the classes S_1 and S_{-1} are not necessarily linearly separable. That is why, we will adapt the constraints of (P_{HM}) to allow some points to be misclassified. This new problem is known as soft margin SVM problem and we will denote it by (P_{SM}). We will prove the existence of a global solution for this problem and give an example for the non uniqueness of global solution. Finally, we will obtain the dual problem of (P_{SM}), which we will denote by (D_{SM}).

- **Chapter 4**. As in the previous chapter we will assume that the classes are not necessarily linearly separable. The idea of this chapter is to transform the data set in such a way that when solving the dual problem (D_{SM}) with the transformed data set we may obtain fewer misclassified points than using the initial data set. After giving an illustrative example we will introduce the kernel functions, which will be of great help in this framework.

- **Chapter 5**. We will use functions from Scikit-learn library in Python, which internally solve the dual problem presented in Chapter 4. We will use the "Breast Cancer Wisconsin (Diagnostic) Data Set" from [7].

Although the main experimental part is covered in Chapter 5, in the previous chapters we will work with some numerical examples using Matlab.

Throughout this paper the author has tried to formulate the problems in a clear way from the mathematical point of view, as well as to prove theoretical results in a rigorous but readable way.

Chapter 2

Hard Margin SVM

2.1 Linearly Separable Data Set

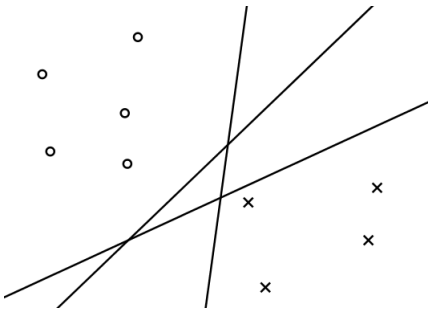

Figure 2.1: Linearly separable data set concept. Class S_1 is represented by $\circ$ and class S_{-1} by $\times$.

Consider a binary data set $S = \{(x_i, y_i) : x_i \in \mathbb{R}^n, y_i \in \{-1, 1\}, i = 1, ..., m\}$ and the classes

$$S_1 = \{x_i : (x_i, 1) \in S\} \text{ and } S_{-1} = \{x_i : (x_i, -1) \in S\}.$$

Remember that we assume that $S_1 \cap S_{-1} = \emptyset$, $S_1 \neq \emptyset$ and $S_{-1} \neq \emptyset$. Then, S_1 and S_{-1} are **linearly separable** if there are $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ such the linear classifier $H : \mathbb{R}^n \to \mathbb{R}$ defined by $H(x) = w^T x + b$ verifies:

$$H(x_i) > 0, \forall\ x_i \in S_1 \text{ and } H(x_j) < 0, \forall\ x_j \in S_{-1}. \tag{2.1}$$

The hyperplane $w^T x + b = 0$ is known as **separating hyperplane.**

We will need the concept of distance between a hyperplane and a point. The following theorem states the well known formula to calculate this value. We present a proof using some optimization results that we have studied last year in the Mathematics Degree.

Theorem 2.1.1. *Given $\hat{x} \in \mathbb{R}^n$ and the hyperplane π defined by $w^T x + b = 0$, then the distance between them is:*

$$dist(\hat{x}, \pi) = \frac{|w^T \hat{x} + b|}{\|w\|_2},$$

where $\|w\|_2 := \sqrt{w^T w}$ represents the 2-norm.

Proof. The distance between $\hat{x}$ and π can be computed using the following nonlinear optimization problem:

$$\begin{cases} min & \sqrt{(x-\hat{x})^T(x-\hat{x})} \\ \\ \text{subject to} & x \in \mathbb{R}^n, \\ & w^T x + b = 0. \end{cases}$$

Notice that we can consider the following equivalent quadratic optimization problem:

$$\begin{cases} min & \frac{1}{2} x^T (2Id_n) x - 2\hat{x}^T x + \hat{x}^T \hat{x} \\ \\ \text{subject to} & x \in \mathbb{R}^n, \\ & w^T x + b = 0. \end{cases}$$

Now, we will use some results from the Appendix. Theorem A.1.2 guarantees the existence of solution for the last problem. Using the Lagrange multipliers rule (Theorem A.1.4) and Corollary A.1.1 it is obtained that:

$$\left.\begin{array}{ll} (i) & 2x - 2\hat{x} + \lambda w = 0, \\ (ii) & w^T x + b = 0. \end{array}\right\}$$

Multiplying equation (i) by w^T, and using (ii) it is obtained that $\lambda = 2\frac{w^T \hat{x} + b}{w^T w}$. Using now this value of λ and (i) we get $x - \hat{x} = -\frac{w^T \hat{x} + b}{w^T w} w$ and, substituting that value in the objective function of the first optimization problem, the proof is completed. □

In computational practice, we want a separation between the two classes S_1 and S_{-1} sharper than the one represented in (2.1). Therefore, we propose the following theorem.

Theorem 2.1.2. *Given $S = \{(x_i, y_i) : x_i \in \mathbb{R}^n, y_i \in \{-1, 1\}, i = 1, ..., m\}$ a separable binary data set and a hyperplane $w^T x + b = 0$ that separates S_1 and S_{-1}, i.e.*

$$w^T x_i + b > 0 \quad \textit{if} \quad x_i \in S_1 \textit{ and } w^T x_j + b < 0 \quad \textit{if} \quad x_j \in S_{-1},$$

then there are $\hat{w} \in \mathbb{R}^n$ *and* $\hat{b} \in \mathbb{R}$ *such that:*

$$\hat{w}^T x_i + \hat{b} \geq 1, \quad \forall\ x_i \in S_1, \tag{2.2}$$

$$\hat{w}^T x_j + \hat{b} \leq -1, \quad \forall\ x_j \in S_{-1}. \tag{2.3}$$

Proof. Let be $x_1 \in S_1$ and $x_{-1} \in S_{-1}$ points with minimum distance to the hyperplane. Notice that $d_1 = w^T x_1 + b > 0$, $d_{-1} = w^T x_{-1} + b < 0$, so it is clear that $d = min\{d_1, -d_{-1}\} > 0$. Let see us that $\hat{w} = w/d$ and $\hat{b} = b/d$ verify (2.2) and (2.3)

- If $x_i \in S_1$, then $\hat{w}^T x_i + \hat{b} = \dfrac{w^T x_i + b}{d} \geq \dfrac{w^T x_1 + b}{d} = \dfrac{d_1}{d} \geq 1.$
- If $x_j \in S_{-1}$, then $\hat{w}^T x_j + \hat{b} = -\dfrac{|w^T x_j + b|}{d} \leq -\dfrac{|w^T x_{-1} + b|}{d} = \dfrac{d_{-1}}{d} \leq -1.$

□

Definition 1 (**Separation margin**). *Given a binary data set and a separating hyperplane* $w^T x + b = 0$*, then the separation margin is the distance between the two hyperplanes defined by* $w^T x + b = 1$ *and* $w^T x + b = -1$*.*

Proposition 2.1.1. *The value of the separation margin is* $2/\|w\|_2$*.*

Proof. It follows from Theorem 2.1.1. □

Remark 2.1.1. *The purpose is to develop a procedure that can be used in practice to find hyperplanes that separates our data set. So, we need to impose some constraints, known as* ***separability constraints****, that arise from inequalities* (2.2) *and* (2.3)*, and they are* $(w^T x_i + b)y_i \geq 1$*, where* w *and* b *are variables.*

2.2 Hard Margin SVM Problem

2.2.1 Constructing the Problem

As we can see in Figure 2.1, if the binary data set is linearly separable there are many hyperplane separating the classes S_1 and S_{-1}. The criteria to determine the best hyperplanes will be the one that maximize the separation margin, and taking into acount Remark 2.1.1, the resultant optimization problem is

$$\begin{cases} max & \hat{f}_{HM}(w,b) = \dfrac{2}{\|w\|_2} \\ \\ \text{subject to} & w \in \mathbb{R}^n,\ b \in \mathbb{R}, \\ & (w^T x_i + b)y_i \geq 1,\ \text{for } i = 1, ..., m. \end{cases} \tag{2.4}$$

We can reformulate the previous non-linear optimization problem to get a convex quadratic optimization problem, that is more convenient to solve,

$$(P_{HM}) \begin{cases} min & f_{HM}(w,b) = \frac{1}{2}w^T w \\ \text{subject to} & w \in \mathbb{R}^n,\ b \in \mathbb{R}, \\ & (w^T x_i + b)y_i \geq 1, \quad \text{for } i = 1, ..., m. \end{cases}$$

Notice that $\binom{\overline{w}}{\overline{b}}$ is solution of problem (2.4) if and only if it is solution of (P_{HM}). Problem (P_{HM}) is known as **hard margin SVM** problem, and it is the optimization problem that determines the parameters $\overline{w}, \overline{b}$ from the separation hyperplanes [1] for a separable binary data set.

2.2.2 Existence and Uniqueness of Solution

Notation: from now on, to simplify notation, in the rest of the paper, only when we will refer to problem solutions, we will write $(\overline{w},\ \overline{b})$ with the meaning of $\binom{\overline{w}}{\overline{b}}$.

Theorem 2.2.1 (Existence of solution for (P_{HM}))**.** *There is at least one solution* $(\overline{w},\ \overline{b})$ *for hard margin SVM problem.*

Proof. Let $\{(w_n, b_n)\}_{n\in\mathbb{N}}$ be a minimizing sequence, that is to say

$$\{(w_n, b_n)\}_{n\in\mathbb{N}} \subset K \text{ and } \lim_{n\to\infty} f_{HM}(w_n, b_n) = inf\{f_{HM}(w,b) :\ (w,b) \in K\}, \qquad (2.5)$$

where K represents the feasible set of problem (P_{HM}). To simplify the notation, we define $\gamma := inf\{f_{HM}(w,b) :\ (w,b) \in K\}$.

We can consider two cases:

- CASE I. The sequence $\{(w_n, b_n)\}_{n\in\mathbb{N}}$ is bounded. So, there is a convergent subsequence $\{(w_{n'}, b_{n'})\}_{n'\in\mathbb{N}} \subset \{(w_n, b_n)\}_{n\in\mathbb{N}}$, and $(w_{n'}, b_{n'}) \xrightarrow{n'\to+\infty} (\overline{w}, \overline{b})$. As K is a closed set, $(\overline{w}, \overline{b}) \in K$. Since f_{HM} is continuous, $f_{HM}(\overline{w}, \overline{b}) = \lim_{n'\to+\infty} f_{HM}(w_{n'}, b_{n'})$. Moreover, $\lim_{n'\to+\infty} f_{HM}(w_{n'}, b_{n'}) = \lim_{n\to+\infty} f_{HM}(w_n, b_n) = \gamma$. Then, $f_{HM}(\overline{w}, \overline{b}) = \gamma$, and it follows that $(\overline{w}, \overline{b})$, is a global solution for (P_{HM}).

- CASE II. The sequence $\{(w_n, b_n)\}_{n\in\mathbb{N}}$ is not bounded. Therefore, there is a subsequence $\{(w_{n'}, b_{n'})\}_{n'\in\mathbb{N}} \subset \{(w_n, b_n)\}_{n\in\mathbb{N}}$ that $\|(w_{n'}, b_{n'})\| \xrightarrow{n'\to+\infty} +\infty$. We distinguish the following two cases:

[1] Some authors refer to these hyperplanes as maximal margin hyperplanes.

- CASE II.a. The subsequence $\{b_{n'}\}_{n'\in\mathbb{N}}$ is bounded, therefore we get that $\|w_{n'}\| \xrightarrow{n'\to+\infty} +\infty$. Taking the 2-norm, we get that $\|w_{n'}\|_2^2 \xrightarrow{n'\to+\infty} +\infty$, and then $f_{HM}(w_{n'}, b_{n'}) = \frac{1}{2}\|w_{n'}\|_2^2 \xrightarrow{n'\to+\infty} +\infty$, but this is an absurd conclusion, because using (2.5) it follows that $f_{HM}(w_{n'}, b_{n'}) \xrightarrow{n'\to+\infty} \gamma$, but $\gamma < +\infty$ because $K \neq \emptyset$.
- CASE II.b. The subsequence $\{b_{n'}\}_{n'\in\mathbb{N}}$ is not bounded. So, there is a subsequence $\{b_{n''}\}_{n''\in\mathbb{N}} \subset \{b_{n'}\}_{n'\in\mathbb{N}}$ that $|b_{n''}| \xrightarrow{n''\to+\infty} +\infty$. Let us suppose that there is a subsequence $\{b_{n'''}\}_{n'''\in\mathbb{N}} \subset \{b_{n''}\}_{n''\in\mathbb{N}}$ verifying $b_{n'''} > 0$, $\forall n''' \in \mathbb{N}$. We can choose $x_i \in S_{-1}$, and then using the separability constraints and the Cauchy-Schwartz inequality

$$|b_{n'''}| = b_{n'''} \leq -1 - (w_{n'''})^T x_i \leq -1 + \| - w_{n'''}\|\|x_i\|.$$

  Since $|b_{n'''}| \xrightarrow{n'''\to+\infty} +\infty$, then $\|w_{n'''}\| \xrightarrow{n''\to+\infty} +\infty$, but that is absurd (using the same argument as in CASE II.a). Analogously, we obtain an absurd result if the subsequence $\{b_{n'''}\}_{n'''\in\mathbb{N}}$ verifies $b_{n'''} < 0$, $\forall n''' \in \mathbb{N}$, but in this case we must take $x_i \in S_1$.

Therefore, only CASE I can occur. □

Remark 2.2.1. *Let us know that (P_{HM}) is a convex problem, so each local solution for (P_{HM}) is also a global solution (Theorem A.1.3).*

Remark 2.2.2. *If $(\overline{w},\ \overline{b})$ is a global solution for (P_{HM}), it is obvious that $\overline{w} \neq 0$. Otherwise, using the separability constraints with one element of each class, we would obtain that $\overline{b} \geq 1$ and $\overline{b} \leq -1$, which is impossible.*

Definition 2 (**Boundary hyperplanes**). *Let $(\overline{w},\ \overline{b})$ be a global solution for (P_{HM}), then the hyperplanes $\overline{w}^T x + \overline{b} = 1$ and $\overline{w}^T x + \overline{b} = -1$ are known as boundary hyperplanes.*

Definition 3 (**Boundary vector**). *Let $(\overline{w},\ \overline{b})$ be a global solution for (P_{HM}), the vector $x_i \in S_1 \cup S_{-1}$ that satisfies $\overline{w}^T x_i + \overline{b} = 1$ or $\overline{w}^T x_i + \overline{b} = -1$ is known as boundary vector.*

The following proposition states that there is always at least one boundary vector of each class. This result will be fundamental in proving the uniqueness of solution for hard margin SVM problem.

Proposition 2.2.1. *For each solution of Problem (P_{HM}) there is at least one boundary vector for each class S_1 and S_{-1}.*

Proof. Proof by reduction to the absurd. Let $(\overline{w},\ \overline{b})$ be a global solution of Problem (P_{HM}) and, let us suppose that there are not boundary vectors of at least one class.

Let be $x_1 \in S_1$ and $x_{-1} \in S_{-1}$ points with minimum distance to the hyperplane $\overline{w}^T x + \overline{b} = 0$. We can define $d_1 = \overline{w}^T x_1 + \overline{b}$ and $d_{-1} = \overline{w}^T x_{-1} + \overline{b}$, notice that $d_1 \geq 1$ and $d_{-1} \leq -1$. Since there are not boundary vectors of at least one class, it follows that at least one of the previous inequalities is strict, so $d = \dfrac{d_1 - d_{-1}}{2} > 1$. Let see that $\hat{w} = \overline{w}/d$ and $\hat{b} = \left(\overline{b} + d - d_1\right)/d$ is a new feasible point of (P_{HM}):

- If $x_i \in S_1$, then

$$\hat{w}^T x_i + \hat{b} = \frac{\overline{w}^T x_i + \overline{b} + d - d_1}{d} \geq \frac{\overline{w}^T x_1 + \overline{b} + d - d_1}{d} = \frac{d_1 + d - d_1}{d} = 1.$$

- If $x_j \in S_{-1}$, then

$$\hat{w}^T x_j + \hat{b} \leq \frac{\overline{w}^T x_{-1} + \overline{b} + d - d_1}{d} = \frac{d_{-1} + d - d_1}{d} = -1,$$

  where in the last equality we have used the relation $d_{-1} = -2d + d_1$.

And $f_{HM}(\hat{w}, \hat{b}) = \frac{1}{2}\hat{w}^T\hat{w} = \frac{1}{2d^2}\overline{w}^T\overline{w} < \frac{1}{2}\overline{w}^T\overline{w} = f_{HM}(\overline{w}, \overline{b})$, but this is absurd because $(\overline{w},\ \overline{b})$ is a global solution for (P_{HM}). □

Theorem 2.2.2 (Uniqueness of solution for (P_{HM}))**.** *Hard margin SVM problem has an unique solution.*

Proof. Let (w_1, b_1) and (w_2, b_2) be two global solutions for (P_{HM}). The proof is divided in two parts, first we will prove that $w_1 = w_2$, and then we will see that $b_1 = b_2$.

1. Let us suppose that $w_1 \neq w_2$. We define $\hat{w} = \frac{1}{2}(w_1 + w_2)$ and $\hat{b} = \frac{1}{2}(b_1 + b_2)$, notice that $(\hat{w}, \hat{b})$ is a feasible point due to the convexity of the feasible set.

   Moreover, due to f_{HM} is a strictly convex function, and two global solutions have the same objective function value, it follows that

$$f_{HM}(\hat{w}, \hat{b}) = \frac{1}{2}\hat{w}^T\hat{w} = \frac{1}{2}\left\|\frac{1}{2}(w_1 + w_2)\right\|_2^2 < \frac{1}{2}\left(\frac{1}{2}\left(\|w_1\|_2^2 + \|w_2\|_2^2\right)\right) =$$

$$= \frac{1}{2}\left(f_{HM}(w_1, b_1) + f_{HM}(w_2, b_2)\right) = f_{HM}(w_1, b_1).$$

   We have obtained an absurd, because (w_1, b_1) is a global solution for (P_{HM}).

   Then, $w_1 = w_2$ and, therefore, the solutions must be (w_1, b_1) and (w_1, b_2).

2. Without loss of generality let us suppose that $b_1 \geq b_2$. Then, there is $\delta \in \mathbb{R}$, $\delta \geq 0$, such that $b_2 = b_1 - \delta$. Due to (w_1, b_2) is a solution of (P_{HM}), if $x_i \in S_1$ then $w_1^T x_i + b_2 \geq 1$, ergo $w_1^T x_i + b_1 \geq 1 + \delta$.

   As (w_1, b_1) is a global solution for (P_{HM}), applying Proposition 2.2.1, there must be $\hat{x} \in S_1$ verifying $w_1^T \hat{x} + b_1 = 1$. Also $\hat{x}$ must satisfy the last inequality, that is to say $w_1^T \hat{x} + b_1 \geq 1 + \delta$. Combining the two expressions (relative to $\hat{x}$) we get that $\delta = 0$, so $b_1 = b_2$.

□

Remark 2.2.3 (Classifying new points)**.** *The objective of finding the best hyperplane is to be able to predict if a new point, x_{new}, belongs to S_1 or S_{-1} through the* ***classification function****, defined as:*

$$class : \mathbb{R}^n \to \{-1, 1\}, \text{ where } class(x) := sign\left(\overline{w}^T x + \overline{b}\right). \text{ [2]} \tag{2.6}$$

If $class(x_{new}) = 1$ then $x_{new} \in S_1$ and, if $class(x_{new}) = -1$ then $x_{new} \in S_{-1}$.

2.3 Duality for Hard Margin SVM Problem

The main objective of this section is to obtain the dual problem of (P_{HM}). For this purpose we will use Wolfe duality (Theorem A.2.1), but instead of applying it directly, a proposition will be introduced.

Proposition 2.3.1. *Consider the following quadratic optimization problem:*

$$\begin{cases} min & \frac{1}{2} x^T H x + p^T x \\ \\ subject\ to & x \in \mathbb{R}^n, \\ & A^T x \leq v, \end{cases} \tag{2.7}$$

where $H \in \mathbb{R}^{n \times n}$ is a positive semidefinite symmetric matrix, $A \in \mathbb{R}^{n \times n_D}$ and $v \in \mathbb{R}^{n_D}$. Then:

1. *If $\overline{x} \in \mathbb{R}^n$ is a global solution of problem (2.7), then there is $\overline{\mu} \in \mathbb{R}^m$ such that $(\overline{x}, \overline{\mu})$ is solution of the Wolfe dual problem:*

$$\begin{cases} max & -\frac{1}{2} x^T H x - v^T \mu \\ \\ subject\ to & (x, \mu) \in \mathbb{R}^n \times \mathbb{R}^{n_D}, \\ & Hx + A\mu = -p, \\ & \mu \geq 0. \end{cases} \tag{2.8}$$

   Also, primal and dual optimal function values are equal.

[2] If $z \in \mathbb{R}$, then $sign(z) = \begin{cases} 1 & if \quad z \geq 0 \\ -1 & if \quad z < 0 \end{cases}$.

2. The dual problem of (2.8) *is*

$$\begin{cases} max & \frac{1}{2}x^T Hx + v^T\mu + (Hx + A\mu + p)^T\lambda - \mu^T\gamma \\ \\ subject\ to & (x, \mu, \lambda, \gamma) \in \mathbb{R}^n \times \mathbb{R}^{n_D} \times \mathbb{R}^n \times \mathbb{R}^{n_D}, \\ & Hx + H\lambda = 0, \\ & v + A^T\lambda - \gamma = 0, \\ & \gamma \geq 0, \end{cases} \tag{2.9}$$

which is equivalent to problem (2.7).

Proof. 1. The Lagrangian function of problem (2.7) is:

$$L(x, \mu) = \frac{1}{2}x^T Hx + p^T x + \mu^T(A^T x - v).$$

Then, $\nabla_x L(x, \lambda, \mu) = Hx + p + A\mu$. To conclude, let us notice that the expression of the Lagrangian function is reduced using the constraint $\nabla_x L(x, \lambda, \mu) = 0$.

Also, Theorem A.2.1 guarantees that the optimal function values are equal. see Theorem A.1.3.

2. Problem (2.8) is equivalent to:

$$\begin{cases} min & \frac{1}{2}x^T Hx + v^T\mu \\ \\ \text{subject to} & (x, \mu) \in \mathbb{R}^n \times \mathbb{R}^{n_D}, \\ & Hx + A\mu = -p, \\ & \mu \geq 0. \end{cases} .$$

Its Lagrangian function is

$$\hat{L}(x, \mu, \lambda, \gamma) = \frac{1}{2}x^T Hx + v^T\mu + (Hx + A\mu + p)^T\lambda - \mu^T\gamma.$$

Applying Wolfe Duality (Theorem A.2.1), using $\nabla_x \hat{L}(x, \mu, \lambda, \gamma) = Hx + H\lambda$ and $\nabla_\mu \hat{L}(x, \mu, \lambda, \gamma) = v + A^T\lambda - \gamma$, we get problem (2.9).

As in the previous case, we can reduce the formula of the Lagrangian function using the constraints, and the resultant problem is:

$$\begin{cases} max & -\frac{1}{2}\lambda^T H\lambda + p^T\lambda \\ \\ \text{subject to} & (\lambda, \gamma) \in \mathbb{R}^n \times \mathbb{R}^{n_D}, \\ & -A^T\lambda + \gamma = v, \\ & \gamma \geq 0. \end{cases}$$

And, using the fact that $\gamma \geq 0$, we get that

$$\begin{cases} max & -\frac{1}{2}\lambda^T H\lambda + p^T\lambda \\ \\ \text{subject to} & \lambda \in \mathbb{R}^n, \\ & -A^T\lambda \leq v. \end{cases}$$

Which is equivalent to problem (2.7) using the change of variable $x = -\lambda$.

□

Remark 2.3.1 (Matrix formulation of (P_{HM}))**.** *In order to apply the previous proposition, notice that* (P_{HM}) *can be rewritten as:*

$$\begin{cases} min & f_{HM}(w,b) = \frac{1}{2}w^T w \\ \\ subject\ to & w \in \mathbb{R}^n,\ b \in \mathbb{R}, \\ & -YX^T w - yb \leq -e, \end{cases} \tag{2.10}$$

where $y = (y_1, ..., y_m)^T$, $e = \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix}^T \in \mathbb{R}^m$, $Y = diag(y) \in \mathbb{R}^{m\times m}$ [3] *and* $X \in \mathbb{R}^{n\times m}$ *is the matrix whose columns are the vectors* x_i.

Proposition 2.3.2. *The Wolfe dual problem of* (2.10) *is:*

$$(\widehat{D_{HM}}) \begin{cases} max & \hat{g}_{HM}(w,\mu) = -\frac{1}{2}w^T w + e^T\mu \\ \\ subject\ to & (w,\mu) \in \mathbb{R}^n \times \mathbb{R}^m, \\ & y^T\mu = 0, \\ & w = XY\mu, \\ & \mu \geq 0. \end{cases}$$

Or, equivalently

$$(D_{HM}) \begin{cases} min & g_{HM}(\mu) = \frac{1}{2}\mu^T YX^T XY\mu - e^T\mu \\ \\ subject\ to & \mu \in \mathbb{R}^m, \\ & y^T\mu = 0, \\ & \mu \geq 0. \end{cases}$$

Proof. Using in Proposition (2.3.1):

$$x = \begin{pmatrix} w \\ b \end{pmatrix} \in \mathbb{R}^{n+1},\ A = \begin{pmatrix} -XY \\ -y^T \end{pmatrix} \in \mathbb{R}^{(n+1)\times m},$$

[3] $Y = diag(y)$ means a diagonal matrix whose diagonal is the vector y.

$$v = \begin{pmatrix} -1 \\ \vdots \\ -1 \end{pmatrix} \in \mathbb{R}^{n+1},\ p = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{n+1} \text{ and } H = \left(\begin{array}{c|c} Id_n & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ \hline 0 \cdots 0 & 0 \end{array}\right) \in \mathbb{R}^{(n+1)\times(n+1)},$$

we obtain that the dual problem of (P_{HM}) is $(\widehat{D_{HM}})$. Moreover, notice that we can rewrite that problem using the constraint

$$w = XY\mu \tag{2.11}$$

in the objective function, and therefore we get (D_{HM}). □

Remark 2.3.2. *Notice that (D_{HM}) is a convex optimization problem, because it is a quadratic programming problem whose Hessian matrix can be seen as M^TM, being M a matrix, so this Hessian matrix is a positive semidefinite symmetric matrix.*

Primal problem (P_{HM}) has m general constraints (separability constraints) while the dual problem (D_{HM}) has m bound constraints and one equality constraint, therefore in practice it is easier to solve (D_{HM}) than (P_{HM}).

In contrast to (P_{HM}), we are not assured of the uniqueness of the solution for (D_{HM}), as we can see in the following example. It is based on a problem presented in [13], but we have adapted it to the hard margin dual case.

Example 2.3.1 (Non uniqueness of solution for (D_{HM}))**.** We consider in $\mathbb{R}^2$ the data points:

$$x_1 = (1,\ 1)^T,\ x_2 = (1,\ 0)^T,\ x_3 = (0,\ 1)^T \text{ and } x_4 = (0,\ 0)^T,$$

with the classification $x_1, x_3 \in S_1$ and $x_2, x_4 \in S_{-1}$. With these data points (D_{HM}) becomes

$$\begin{cases} min & g_{HM}(\mu) = \dfrac{1}{2}(2\mu_1^2 + \mu_2^2 + \mu_3^2 - 2\mu_1\mu_2 + 2\mu_1\mu_3) - \displaystyle\sum_{i=1}^{4} \mu_i \\ \text{subject to} & \mu_i \in \mathbb{R}, \forall i = 1, ..., 4, \\ & \mu_1 - \mu_2 + \mu_3 - \mu_4 = 0, \\ & -\mu_1 \leq 0,\ -\mu_2 \leq 0,\ -\mu_3 \leq 0,\ -\mu_4 \leq 0. \end{cases}$$

As the previous problem is convex (Remark 2.3.2), every Kuhn-Tucker point is a global solution (Theorem A.1.5). The Kuhn-Tucker points (see Appendix A.1.1) must verify the constraints of the problem and also the equations:

$$\begin{pmatrix} 2\mu_1 - \mu_2 + \mu_3 - 1 \\ \mu_2 - \mu_1 - 1 \\ \mu_3 + \mu_1 - 1 \\ -1 \end{pmatrix} + \lambda \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} + \gamma_1 \begin{pmatrix} -1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \gamma_2 \begin{pmatrix} 0 \\ -1 \\ 0 \\ 0 \end{pmatrix} + \gamma_3 \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \end{pmatrix} + \gamma_4 \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$\gamma_1(-\mu_1) = 0,\ \ \gamma_2(-\mu_2) = 0,\ \ \gamma_3(-\mu_3) = 0,\ \ \gamma_4(-\mu_4) = 0,$$

where $\lambda \in \mathbb{R}$ and $\gamma_i \in \mathbb{R}, \gamma_i \geq 0,\ \forall\, i = 1, ..., 4$.

It is easy to prove that $\overline{\mu} = (1, 1, 1, 1)^T$ satisfies the constraints of the problem, and it also verifies the previous equations with the Lagrange multipliers $\overline{\lambda} = -1$, $\overline{\gamma}_1 = 0$, $\overline{\gamma}_2 = 0, \overline{\gamma}_3 = 0$ and $\overline{\gamma}_4 = 0$. So, $\overline{\mu}$ is a global solution whose objective function value is $g_{HM}(\overline{\mu}) = -2$.

Furthermore, $g_{HM}(t, t, 2-t, 2-t) = -2,\ \forall t \in [0,\ 2]$, and therefore, we get that the feasible vectors $(t, t, 2-t, 2-t)^T,\ \forall t \in [0,\ 2]$, are also global solutions.

Notice, that every global solution of the dual problem (D_{HM}) will generate the first n components of the unique solution of the primal problem (P_{HM}). This is guaranteed by the following proposition.

Proposition 2.3.3. *Let $\overline{\mu}$ be a global solution for (D_{HM}), then the coordinates of $\overline{w} = XY\overline{\mu}$ are the first n components of the unique solution of the primal problem (P_{HM}).*

Proof. If $\overline{\mu}$ is a global solution for (D_{HM}), we can consider $(\overline{w}, \overline{\mu})$ as a solution of $(\widehat{D_{HM}})$. Due to the proof of Proposition 2.3.1, there is $\lambda^* \in \mathbb{R}^{n+1}$ such that $-\lambda^*$ is the solution of the primal problem (P_{HM}). Moreover, using the constraint $Hx + H\lambda = 0$ from (2.8), where H and x are the same as in the proof of Proposition 2.3.2, it follows that $-\lambda_i^* = \overline{w}_i$ for $i = 1, ..., n$. □

Definition 4 (Support vector). *Let $\overline{\mu}$ be a global solution for (D_{HM}), then the data point $x_i \in\ S_1 \cup S_{-1}$ for which $\overline{\mu}_i > 0$ is known as support vector.*

Notice that $\overline{w}$ is written as a lineal combination of support vectors (remember formula (2.11)), therefore, they influence the classification of new points. And this is why this machine learning procedure for data classification is called Support Vector Machine, because it only relies on support vectors the classification of new points.

Proposition 2.3.4. *Let $\overline{\mu}$ be a solution of the dual problem (D_{HM}). If $\overline{\mu}_i > 0$, then the corresponding separability constraint is active at $(\overline{w}, \overline{b})$, i.e.*

$$\left(\overline{w}^T x_i + \overline{b}\right) y_i = 1, \tag{2.12}$$

where $(\overline{w},\ \overline{b})$ is the solution of the primal problem (P_{HM}).

Proof. Let us suppose that there is $i_0 \in \{1, ..., m\}$ such that

$$\left(\overline{w}^T x_{i_0} + \overline{b}\right) y_{i_0} > 1 \text{ and } \overline{\mu}_{i_0} > 0.$$

So, we get $\overline{\mu}_{i_0} \left(\overline{w}^T x_{i_0} + \overline{b}\right) y_{i_0} > \overline{\mu}_{i_0}$. Moreover, taking into account the constraints of (P_{HM}) and (D_{HM}):

$$(\overline{w}^T x_i + \overline{b}) y_i \geq 1, \forall i = 1, ..., m,$$

$$y^T\overline{\mu} = 0, \quad \overline{w} = XY\overline{\mu} \ \text{ and } \ \overline{\mu} \geq 0,$$

we obtain that $\sum_{\overline{\mu}_i>0} \overline{\mu}_i \left(\overline{w}^T x_i + \overline{b}\right) y_i > \sum_{\overline{\mu}_i>0} \overline{\mu}_i$. Furthermore, as $y^T\overline{\mu} = 0$, we get that $\sum_{\overline{\mu}_i>0} \overline{\mu}_i \overline{w}^T x_i y_i > e^T\overline{\mu}$, and using that $\overline{w} = XY\overline{\mu}$, it follows that $\overline{w}^T XY\overline{\mu} > e^T\overline{\mu}$, which is equivalent to

$$\tfrac{1}{2}\overline{w}^T\overline{w} > -\tfrac{1}{2}\overline{\mu}^T Y X^T XY\overline{\mu} + e^T\overline{\mu}.$$

We have obtained that the optimal function value of the dual problem is strictly less than the primal one, which is absurd by Wolfe Duality (Theorem A.2.1). □

The previous proposition states that all support vectors are boundary vectors, but the other implication is not true, see Example 2.3.3. But first, we must consider the following remark.

Remark 2.3.3 (Calculating the hyperplane after solving (D_{HM}))**.** *Let $\overline{\mu}$ be a global solution for (D_{HM}), due to Proposition 2.3.3 we have that $\overline{w} = XY\overline{\mu}$. Notice that $\overline{b}$ can be found using the constraints of (P_{HM}), with one of the following strategies:*

1. *Using the formula* (2.12)*, with a support vector x_i, i.e.*

$$\overline{b} = y_i - \overline{w}^T x_i. \tag{2.13}$$

2. *Using in the previous item a support vector whose corresponding $\overline{\mu}_p$ is sufficiently dominant over the rest of $\overline{\mu}_i$.*

3. *As $(\overline{w},\ \overline{b})$ is the solution for (P_{HM}), it must verify the separability constraints constraints, ergo*

   - $\overline{b} \geq 1 - \overline{w}^T x_i \ \ \forall x_i \in S_1$*, then*

   $$\overline{b} \geq \max_{x_i \in S_1}(1 - \overline{w}^T x_i) = 1 - \min_{x_i \in S_1}(\overline{w}^T x_i).$$

   - $\overline{b} \leq -1 - \overline{w}^T x_j \ \ \forall x_j \in S_{-1}$*, then*

   $$\overline{b} \leq \min_{x_j \in S_{-1}}(-1 - \overline{w}^T x_j) = -1 - \max_{x_j \in S_{-1}}(\overline{w}^T x_j).$$

   Then, we can choose the medium point of the interval defined by the above bounds (see [6])

$$\overline{b} = -\frac{\min_{x_i \in S_1}(\overline{w}^T x_i) + \max_{x_j \in S_{-1}}(\overline{w}^T x_j)}{2}. \tag{2.14}$$

Notice that we can also calculate $\overline{b}$ *with the formula* (2.13) *for each support vector and then compute the mean, but we will have to choose a tolerance to decide which vectors are support vectors and which are not. This is why we propose the formula* (2.14)*, in order not to use tolerances.*

Example 2.3.2 (Numerical experiments, strategies for calculating $\overline{b}$)**.** We consider in $\mathbb{R}^2$ the data points in Table 2.1, with the classification $x_1, x_4, x_5, x_6, x_7 \in S_1$ and $x_2, x_3, x_8, x_9 \in S_{-1}$.

$x_1 = (2,\ 2)^T$
$x_2 = (-1,\ 1)^T$
$x_3 = (1,\ 4)^T$
$x_4 = (4,\ 1)^T$
$x_5 = (6,\ 1)^T$
$x_6 = (1,\ 0)^T$
$x_7 = (5,\ 4)^T$
$x_8 = (2,\ 5)^T$
$x_9 = (-2,\ 3)^T$

Table 2.1: Data set of Example 2.3.2.

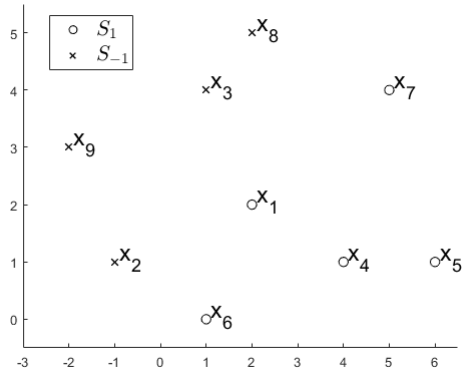

Figure 2.2: Data representation of Example 2.3.2.

Using *quadprog*, a quadratic programming solver provided in Matlab's Optimization Toolbox, a numerical solution for (D_{HM}) with the previous data is represented in the first column of Table 2.2, therefore $\overline{w} = XY\overline{\mu} = (8.8889e-01,\ -6.6667e-01)^T$. The second column of Table 2.2 represents the value of $\overline{b}$ for each data point calculated using the formula $\overline{b}_{x_i} = y_i - \overline{w}^T x_i$.

$\overline{\mu}_i$	$\overline{b}_{x_i}$
6.1728e-01	5.5556e-01
2.9630e-01	5.5556e-01
3.9324e-09	7.7778e-01
5.3683e-11	-1.8889e+00
3.5211e-11	-3.6667e+00
5.0960e-10	1.1111e-01
7.2310e-11	-7.7778e-01
3.2099e-01	5.5556e-01
6.8335e-11	2.7778e+00

Table 2.2: $\overline{\mu}$ and $\overline{b}$ of Example 2.3.2.

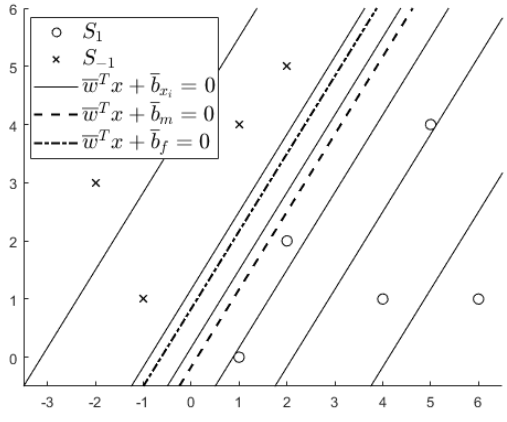

Figure 2.3: Different hyperplanes according to the value of $\overline{b}$ chosen.

Figure 2.3 shows the different hyperplanes according to the value of $\overline{b}$ chosen, where $\overline{b}_m = -1.1111e{-}01$ represents the mean of the values $\overline{b}_{x_i}$, and $\overline{b}_f = 5.5556e{-}01$ is the value computed using the formula (2.14).

Notice that if we take as tolerance $1.0e-12$ all the data points will be support vectors. And, if we use the first strategy of Remark 2.3.3, depending on which one we choose to compute $\overline{b}$ the resulting hyperplane may not separate the data set, see Figure 2.3. If we use the same tolerance, and calculate the average of the values $\overline{b}_{x_i}$, i.e. $\overline{b}_m$, we can observe how the resultant hyperplane is not at the same distance from the class S_1 to the class S_{-1}, which is not the case if we use the value $\overline{b}_f$.

We conclude this section by showing an example that demonstrates that not every boundary vector is a support vector, and therefore it makes sense to define both concepts. The idea of the following example has been extracted from [1].

Example 2.3.3. We consider in $\mathbb{R}^2$ the data points: $x_1 = (0,0)^T$, $x_2 = (1,0)^T$ and $x_3 = (0,1)^T$, with the classification $x_1, x_2 \in S_1$ and $x_3 \in S_{-1}$. (D_{HM}) with this data is equivalent to

$$\begin{cases} min & g_{HM}(\mu) = \dfrac{1}{2}(\mu_2^2 + \mu_3^2) - \mu_1 - \mu_2 - \mu_3 \\ \\ \text{subject to} & \mu_1, \mu_2, \mu_3 \in \mathbb{R}, \\ & \mu_1 + \mu_2 - \mu_3 = 0, \\ & -\mu_1 \leq 0, \\ & -\mu_2 \leq 0, \\ & -\mu_3 \leq 0. \end{cases}$$

A Kuhn-Tucker point of the previous problem is $\overline{\mu} = (2,\ 0,\ 2)^T$, whose Lagrange multiplier associated with the equality constraint is $\overline{\lambda} = 1$, and to the bound ones are $\overline{\gamma}_1 = 0, \overline{\gamma}_2 = 0$ and $\overline{\gamma}_3 = 0$. Since the problem is convex (Remark 2.3.2), every Kuhn-Tucker point is a global solution (Theorem A.1.5).

Moreover, using Remark 2.3.3, we obtain $\overline{w} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 2 \\ 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \end{pmatrix}$ and $\overline{b} = 1$. And therefore, x_2 is a boundary vector (because it verifies $(\overline{w}^T x_2 + \overline{b})y_2 = 1$) but not a support vector since $\overline{\mu}_2 = 0$.

Chapter 3

Soft Margin SVM

The aim of this chapter is to consider binary data sets

$$S = \{(x_i, y_i) : x_i \in \mathbb{R}^n, y_i \in \{-1, 1\}, i = 1, ..., m\}$$

that are not necessarily linearly separable.

3.1 Soft Margin SVM Problem

3.1.1 Constructing the Problem

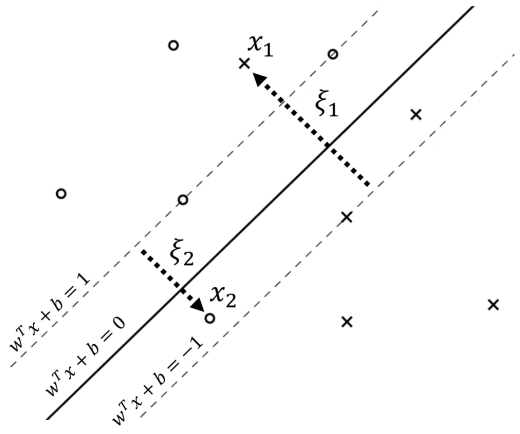

Figure 3.1: Soft margin concept. Class S_1 is represented by $\circ$ and class S_{-1} by $\times$. The data points x_1 and x_2 are missclasified.

When dealing with real-world data, it is common that the data set is not linearly separable, so hard margin SVM problems are not able to handle it. Therefore, we

should relax the separability constraints by adding positive **slack variables** ξ_i, i.e.

$$w^T x_i + b \geq 1 - \xi_i, \text{ for } x_i \in S_1,$$

$$w^T x_j + b \leq -1 + \xi_j, \text{ for } x_j \in S_{-1}.$$

The consequence of relaxing the constraints is to have some data to be misclassified, as is the case of x_1 and x_2 in Figure 3.1.

In order to control the relaxation of the separability constraints, we add a penalty term to the objective function. So, the quadratic optimization problem, known as **soft margin SVM** problem, becomes

$$(P_{SM}) \begin{cases} min & f_{SM}(w,b,\xi) = \frac{1}{2}w^T w + C\sum_{i=1}^{m}\xi_i \\ \text{subject to} & w \in \mathbb{R}^n,\ b \in \mathbb{R},\ \xi \in \mathbb{R}^m, \\ & (w^T x_i + b)y_i \geq 1 - \xi_i, \text{ for } i = 1,...,m, \\ & \xi_i \geq 0, \text{ for } i = 1,...,m, \end{cases}$$

where $C > 0$ is known as the **regularization parameter**. Obviously, the higher C is, the more the violation of the constraints will be penalized.

Therefore, the goal of (P_{SM}) is to find the parameters $\overline{w}$, $\overline{b}$ of some hyperplanes that maximizes the separation margin between the two classes S_1 and S_{-1} while controlling the violation of the separability constraints.

3.1.2 Existence and Non-Uniqueness of Solution

Theorem 3.1.1 (Existence of solution for (P_{SM}))**.** *There is at least one global solution* $\left(\overline{w},\ \overline{b},\ \overline{\xi}\right)$ *for the soft margin SVM problem.*

Proof. It is analogous to the proof of Theorem 2.2.1, which we have done in the previous chapter for (P_{HM}), but taking $w_\xi = \begin{pmatrix} w \\ \xi \end{pmatrix}$ and, instead of the 2-norm using the following fact

$$f_{SM}(w,b,\xi) = \frac{1}{2}w^T w + Ce^T\xi = \frac{1}{2}\|w\|_2^2 + C\|\xi\|_1 \geq M\|w\|_1^2 + C\|\xi\|_1 \geq$$

$$\geq min\{M\|w\|_1, C\}(\|w\|_1 + \|\xi\|_1) = min\{M\|w\|_1, C\}\left\|\begin{pmatrix} w \\ \xi \end{pmatrix}\right\|_1,$$

where $M \in \mathbb{R}, M > 0$ (we have used the fact that all norms in $\mathbb{R}^n$ are equivalent) and $\|\xi\|_1 = \sum_{i=1}^{m}|\xi_i|$ represents the 1-norm. □

Let $\left(\overline{w},\ \overline{b},\ \overline{\xi}\right)$ be a solution of (P_{HM}), it is clear that if $\overline{\xi}_i = 0,\ \forall i = 1, ..., m$, the data set is linearly separable. But if this is not the case, the following remark will help us to understand how to interpret the separability of our data set from the values taken by the slack variables.

Remark 3.1.1 (Interpreting the values $\overline{\xi}_i$)**.** *Let $\left(\overline{w},\ \overline{b},\ \overline{\xi}\right)$ be a global solution for (P_{SM}). Notice that the values $\overline{\xi}_i$ provide some information about how separable our data set is in relation to the hyperplane $\overline{w}^T x + \overline{b} = 0$. Consider $x_i \in S_1 \cup S_{-1}$, we distinguish four cases:*

- *If $\overline{\xi}_i = 0$, then x_i satisfies $(\overline{w}^T x_i + \overline{b})y_i \geq 1$, therefore $class(x_i) = y_i$, and it is well classified.*
- *If $0 < \overline{\xi}_i < 1$, then x_i is between the hyperplanes $\overline{w}^T x + \overline{b} = 0$ and $\overline{w}^T x + \overline{b} = y_i$, therefore $class(x_i) = y_i$, so it is well classified.*
- *If $\overline{\xi}_i = 1$, then x_i verifies $\overline{w}^T x_i + \overline{b} = 0$, therefore $class(x_i) = 1$ and it may be misclassified.*
- *If $\overline{\xi}_i > 1$, it follows that $class(x_i) = -y_i$, so it is misclassified. And, we can interpret this fact as the point x_i is not separable with respect to the hyperplane.*

Clearly, the more $\overline{\xi}_i > 1$ cases there are, the less effectively the hyperplane will separate the data, resulting in a greater number of misclassified data points and therefore a poorer quality of the hyperplane.

In practice, it is common to try different values of C until the desired result is obtained, i.e. the hyperplane with the lowest number of misclassified points. So, we present an example in which we can see how the missclasified points and the expression of the hyperplane varies depending on the value of the regularization parameter.

Example 3.1.1 (Experimenting with C)**.** Consider in $\mathbb{R}^2$ the non linearly separable data points:

$$x_1 = (1,\ 1)^T,\ x_2 = (1.5,\ 2)^T,\ x_3 = (3,\ 4)^T,\ x_4 = (4,\ 4)^T,\ x_5 = (2,\ 1)^T,$$

$$x_6 = (3,\ 2)^T,\ x_7 = (4,\ 1)^T,\ x_8 = (0.7,\ 3)^T,\ x_9 = (2.5,\ 3)^T \text{ and } x_{10} = (3.5,\ 3)^T,$$

with the classification: $x_1, x_2, x_3, x_4, x_9 \in S_1$ and $x_5, x_6, x_7, x_8, x_{10} \in S_{-1}$.

In Table 3.1 values of the slack variables are shown (rounded to 2 decimal digits and we have written 0 in the case that $\overline{\xi}_i$ is less than 10^{-10}) for some values of C after solving (P_{SM}) with the solver *quadprog* from Matlab.

$\overline{\xi}$	$C = 0.1$	$C = 0.5$	$C = 2$	$C = 10$
$\overline{\xi}_1$	1.40	1.58	1.20	0.67
$\overline{\xi}_2$	1.06	1.03	0.40	0
$\overline{\xi}_3$	0.44	0	0	0
$\overline{\xi}_4$	0.56	0.14	0	0.67
$\overline{\xi}_5$	0.48	0.28	0	0

$\overline{\xi}$	$C = 0.1$	$C = 0.5$	$C = 2$	$C = 10$
$\overline{\xi}_6$	0.76	0.76	0.40	0
$\overline{\xi}_7$	0.24	0	0	0
$\overline{\xi}_8$	1.43	1.70	3.44	4.40
$\overline{\xi}_9$	0.78	0.55	0	0
$\overline{\xi}_{10}$	1.10	1.31	1.20	0.67

Table 3.1: Example 3.1.1, values of $\overline{\xi}$ depending on C.

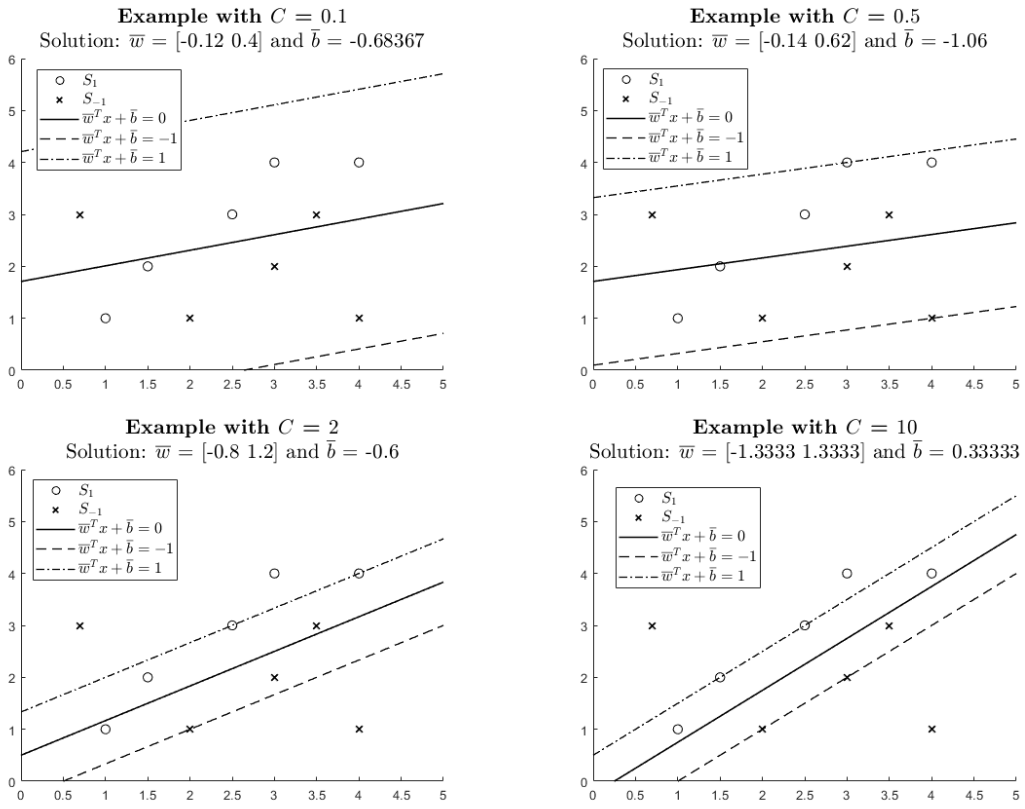

Figure 3.2: Results of Example 3.1.1.

Also, we can see in Table 3.1 that as C gets bigger the number of $\overline{\xi}_i$ verifying $\overline{\xi}_i = 0$ increases, and that is because bigger values of C penalize the violation of the constraints.

In the graphs of Figure 3.2, we can observe how the hyperplane considerably changes and the variation of missclasified points as the value of C varies. We get the best result with $C = 10$, and for larger values of C we still get the same hyperplane. Additionally, in the upper left graph of Figure 3.2, we can see that there are not

boundary vectors for either of the two classes, something that did not occur with Hard-Margin SVM problems.

Remark 3.1.2. *By generalising the Hard-Margin SVM problem, we have preserved the convexity of the problem, but not the uniqueness of solution.*

Example 3.1.2 (Non uniqueness of solution for (P_{SM}), based on [3])**.** We consider in $\mathbb{R}$ the data points $x_1 = 1$ and $x_2 = -1$, with the classification $x_1 \in S_1$ and $x_2 \in S_{-1}$. If the regularization parameter is $C = 0.25$, then (P_{SM}) with this data set is equivalent to

$$\begin{cases} min & f_{SM}(w,b,\xi) = \frac{1}{2}w^2 + 0.25(\xi_1+\xi_2) \\ \\ \text{subject to} & w \in \mathbb{R},\ b \in \mathbb{R},\ , \xi_1 \in \mathbb{R},\ \xi_2 \in \mathbb{R}, \\ & -w - b + 1 - \xi_1 \leq 0, \\ & -w + b + 1 - \xi_2 \leq 0, \\ & -\xi_1 \leq 0, \\ & -\xi_2 \leq 0. \end{cases}$$

Any Kuhn-Tucker point (see Section A.1.1) of the previous problem must verify its constraints and also the following equations:

$$\begin{pmatrix} w \\ 0 \\ 0.25 \\ 0.25 \end{pmatrix} + \mu_1 \begin{pmatrix} -1 \\ -1 \\ -1 \\ 0 \end{pmatrix} + \mu_2 \begin{pmatrix} -1 \\ 1 \\ 0 \\ -1 \end{pmatrix} + \mu_3 \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \end{pmatrix} + \mu_4 \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$\mu_1(-w-b+1-\xi_1) = 0, \quad \mu_2(-w+b+1-\xi_2) = 0, \quad \mu_3(-\xi_1) = 0, \quad \mu_4(-\xi_2) = 0,$$

where $\mu_i \in \mathbb{R}, \mu_i \geq 0,\ \forall\, i = 1, ..., 4$.

It is easy to prove that the feasible point $\overline{w} = 0.5,\ \overline{b} = 0,\ \overline{\xi}_1 = 0.5$ and $\overline{\xi}_2 = 0.5$ verifies the previous equations with the Lagrange multipliers $\overline{\mu}_1 = \overline{\mu}_2 = 0.25$ and $\overline{\mu}_3 = \overline{\mu}_4 = 0$. Therefore, as the problem is convex (Remark 3.1.2) we get that $\left(\overline{w},\ \overline{b},\ \overline{\xi}\right)$ is a global solution (Theorem A.1.5), whose optimal objective function value is $f_{SM}\left(\overline{w}, \overline{b}, \overline{\xi}\right) = 3/8$.

By the other hand, the feasible points $\widehat{w} = 0.5,\ \widehat{b} \in [-0.5,\ 0.5],\ \widehat{\xi}_1 = 0.5 - \widehat{b}$ and $\widehat{\xi}_2 = 0.5 + \widehat{b}$ verifies that $f_{SM}(\widehat{w}, \widehat{b}, \widehat{\xi}\ ) = 3/8$, so they are also global solutions.

As we have just seen in the previous example, we have not guaranteed the uniqueness of the solution for (P_{SM}). But we have that all the solution hyperplanes of (P_{SM}) will be parallel, and this is given by the following proposition.

Proposition 3.1.1. *If $(\overline{w}, \overline{b}, \overline{\xi}\ )$ and $(\widehat{w}, \widehat{b}, \widehat{\xi}\ )$ are two different global solutions for (P_{SM}), then $\overline{w} = \widehat{w}$.*

Proof. Let us suppose that $\overline{w} \neq \widehat{w}$. We define $w^* = \frac{1}{2}(\overline{w} + \widehat{w})$, $b^* = \frac{1}{2}\left(\overline{b} + \widehat{b}\right)$ and $\xi^* = \frac{1}{2}\left(\overline{\xi} + \widehat{\xi}\right)$, and notice that (w^*, b^*, ξ^*) is a feasible point for (P_{SM}) due to the convexity of the feasible set.

By the other hand, we have that

$$f_{SM}(w^*, b^*, \xi^*) = \frac{1}{2}\left\|\frac{1}{2}(\overline{w} + \widehat{w})\right\|_2^2 + \frac{C}{2}\sum_{i=1}^{m}\left(\overline{\xi}_i + \widehat{\xi}_i\right) < \frac{1}{2}\left(\frac{1}{2}\left(\|\overline{w}\|_2^2 + \|\widehat{w}\|_2^2\right)\right) +$$

$$+ \frac{C}{2}\sum_{i=1}^{m}\left(\overline{\xi}_i + \widehat{\xi}_i\right) = \frac{1}{2}\left(f_{SM}(\overline{w}, \overline{b}, \overline{\xi}\,) + f_{SM}(\widehat{w}, \widehat{b}, \widehat{\xi}\,)\right) = f_{SM}(\widehat{w}, \widehat{b}, \widehat{\xi}\,),$$

in the last equality we have used that both solutions are global. Therefore, we have obtained that $f_{SM}(w^*, b^*, \xi^*) < f_{SM}(\widehat{w}, \widehat{b}, \widehat{\xi}\,)$, and this is absurd, because $(\widehat{w}, \widehat{b}, \widehat{\xi}\,)$ is a global solution. □

3.2 Degeneracy

Curiously, in contrast to what happened in Hard-Margin SVM problems, the case where $\overline{w} = 0$ is now possible. This is known as **degeneracy**, and the consequence is that all points will be classified into one class. In principle, it does not matter in which class all the data will be classified, because the idea of this method is to find a hyperplane that separates the classes while minimizing the number of misclassified points, and this is not achieved in this way.

Example 3.2.1 (Degeneracy example, based on [1])**.** We consider in $\mathbb{R}$ the points: $x_1 = -1$, $x_2 = 0$ and $x_3 = 1$, with the classification $x_1, x_3 \in S_1$ and $x_2 \in s_{-1}$. With this data set, problem (P_{SM}) becomes

$$\begin{cases} min & f_{SM}(w, b, \xi) = \frac{1}{2}w^2 + C\,(\xi_1 + \xi_2 + \xi_3) \\ \\ \text{subject to} & w \in \mathbb{R},\ b \in \mathbb{R},\ \xi \in \mathbb{R}^3, \\ & w - b + 1 - \xi_1 \leq 0, \\ & b + 1 - \xi_2 \leq 0, \\ & -w - b + 1 - \xi_3 \leq 0, \\ & -\xi_1 \leq 0, \\ & -\xi_2 \leq 0, \\ & -\xi_3 \leq 0. \end{cases} \tag{3.1}$$

A Kuhn-Tucker point of the previous problem is

$$\overline{w} = 0,\ \overline{b} = 1,\ \overline{\xi}_1 = 0,\ \overline{\xi}_2 = 2,\ \overline{\xi}_3 = 0,\ \forall\, C > 0,$$

whose Lagrange multipliers associated with the general constraints are $\overline{\mu}_1 = C/2$, $\overline{\mu}_2 = C$, $\overline{\mu}_3 = C/2$ and $\overline{\mu}_4 = C/2$, $\overline{\mu}_5 = 0$, $\overline{\mu}_6 = C/2$ to the bound ones. As the problem is convex (Remark 3.1.2), the previous Kuhn-Tucker point is a global solution. Therefore, using the classification function (Remark 2.2.3) any point will be classified into S_1.

For more information on degeneracy and when it occurs see [11].

3.3 Duality for Soft Margin SVM Problem

Remark 3.3.1 (Matrix formulation of (P_{SM}))**.** *Notice that we can rewrite* (P_{SM}) *as follows*

$$\begin{cases} \textit{min} & f_{SM}(w,b,\xi) = \frac{1}{2}w^T w + Ce^t\xi \\ \\ \textit{subject to} & w \in \mathbb{R}^n,\ b \in \mathbb{R},\ \xi \in \mathbb{R}^m, \\ & -YX^T w - yb - \xi \leq -e, \\ & -\xi \leq 0, \end{cases} \tag{3.2}$$

where, $y = (y_1, ..., y_m)^T$, $e = \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix}^T \in \mathbb{R}^m$, $Y = diag(y) \in \mathbb{R}^{m\times m}$ *and* $X \in \mathbb{R}^{n\times m}$ *the matrix whose columns are the vectors* x_i .

Proposition 3.3.1. *The Wolfe dual problem of* (3.2) *is*

$$(\widehat{D_{SM}}) \begin{cases} \textit{max} & \hat{g}_{SM}(w,\eta,\nu) = -\frac{1}{2}w^T w + e^T\eta \\ \\ \textit{subject to} & (w,\eta,\nu) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m, \\ & w = XY\eta, \\ & y^T\eta = 0, \\ & \eta_i + \nu_i = C, \quad \textit{for } i = 1, ..., m, \\ & \eta \geq 0, \\ & \nu \geq 0, \end{cases}$$

which is equivalent to

$$(D_{SM}) \begin{cases} \textit{min} & g_{SM}(\eta) = \frac{1}{2}\eta^T YX^T XY\eta - e^T\eta \\ \\ \textit{subject to} & \eta \in \mathbb{R}^m, \\ & y^T\eta = 0, \\ & 0 \leq \eta \leq C. \end{cases}$$

Proof. Let us use in Proposition 2.3.1 the following expressions

$$x = \left(w^T,\ b,\ \xi^T\right)^T \in \mathbb{R}^{n+1+m}, \qquad p = \left(0,\ \overset{(n+1)}{\ldots},\ 0,\ C,\ \overset{(m)}{\ldots},\ C\right)^T \in \mathbb{R}^{n+1+m},$$

$$v = \left(-1,\ \overset{(m)}{\ldots},\ -1,\ 0,\ \overset{(m)}{\ldots},\ 0\right)^T \in \mathbb{R}^{2m},\ \ \mu = \left(\eta_1,\ \ldots,\ \eta_m,\ \nu_1,\ \ldots,\ \nu_m\right)^T \in \mathbb{R}^{2m},$$

$$A = \left(\begin{array}{c|c} \begin{matrix} -XY \\ -y^T \end{matrix} & \mathbf{0}_{n+1,m} \\ \hline -Id_m & -Id_m \end{array}\right) \quad \text{and} \quad H = \left(\begin{array}{cc|c} & 0 & \\ Id_n & \vdots & \mathbf{0}_{n+1,m} \\ & 0 & \\ \hline 0 \cdots 0 & 0 & \\ \hline \multicolumn{2}{c|}{\mathbf{0}_{m,n+1}} & \mathbf{0}_{m,m} \end{array}\right).$$

Now, working with the constraint $Hx + A\mu = -p$, we obtain:

- $w - XY\eta + \mathbf{0}_{m,m}\ \nu = 0$, so $w = XY\eta$.
- $0 \cdot b - y^T\eta = 0$, so $y^T\eta = 0$.
- $\left(-Id_m \quad -Id_m\right)\mu = -C\left(1,\ \overset{(m)}{\ldots},\ 1\right)^T$, so $\eta_i + \nu_i = C$ for $i = 1, ..., m$.

Proposition 2.3.1 gives also the constraints $\eta \geq 0$ and $\nu \geq 0$. The objective function is $-\frac{1}{2}x^T Hx - v^T\mu = -\frac{1}{2}w^T w + e^T\eta$. And, therefore the dual problem is $(\widehat{D_{SM}})$.

In order to get (D_{SM}), we have to combine the constraints $\eta \geq 0$, $\nu \geq 0$ and $\eta_i + \nu_i = C$ to get $0 \leq \eta \leq C$. And, the objective function is reformulated using the constraint $w = XY\eta$. □

It is remarkable the similarity between (D_{SM}) and (D_{HM}). The only difference between them are the upper bound constraints. And, due to this resemblance, we know that (D_{SM}) is a convex optimization problem, and furthermore, the uniqueness of solution for (D_{SM}) is not guaranteed (to verify this fact, it is enough to take $C = 2$ and the data set of the Example 2.3.1).

Remark 3.3.2. *In Table 3.2 we can see the substantial difference between $(\widehat{D_{SM}})$ and (D_{SM}), in terms of the number of variables and constraints. Therefore, in practice, we solve (D_{SM}) and not $(\widehat{D_{SM}})$. Also, in Table 3.2, if we take into account the types of constraints and the number of variables, we can observe the profit of solving (D_{SM}) instead of (P_{SM}).*

Also, notice that if $\overline{\eta}$ is a global solution for (D_{SM}), then $(w^*,\ \overline{\eta},\ \nu^*)$ is a global solution for $(\widehat{D_{SM}})$, where $\nu_i^* = C - \overline{\eta}_i$ for $i = 1, ..., m$, and $w^* = XY\overline{\eta}$. And, obviously $\hat{g}_{SM}(w^*,\ \overline{\eta},\ \nu^*) = g_{SM}(\overline{\eta})$.

	(P_{SM})	$(\widehat{D_{SM}})$	(D_{SM})
Variables	$w,\ b,\ \xi$	$w,\ \eta,\ \nu$	η
Number of variables	$n+1+m$	$n+2m$	m
Number of equality constraints	$-$	$n+1+m$	1
Number of general constraints	m	$-$	$-$
Number of bound constraints	m	$2m$	$2m$

Table 3.2: Comparison between (P_{SM}), $(\widehat{D_{SM}})$ and (D_{SM}).

Proposition 3.3.2. *Let $\overline{\eta}$ be a global solution for the dual problem (D_{SM}), then:*

1. $\overline{\eta} \neq 0$.

2. *The coordinates of $\overline{w} = XY\overline{\eta}$ are the first n components of a solution for (P_{SM}).*

3. *(**Complementary conditions**). For $i = 1, ..., m$, it is verified that*

$$\overline{\eta}_i \left[y_i \left(\overline{w}^T x_i + \overline{b} \right) - 1 + \overline{\xi}_i \right] = 0,$$

$$\overline{\xi}_i \left(C - \overline{\eta}_i \right) = 0,$$

where $\left(\overline{w},\ \overline{b},\ \overline{\xi}\right)$ is a global solution for (P_{SM}) and $\overline{w} = XY\overline{\eta}$.

Proof. Let us prove the proposition item by item.

1. Let us suppose that $\overline{\eta} = 0$, so the dual optimal function value is $g_{SM}(\overline{\eta}) = 0$. Due to Wolfe Duality (Theorem A.2.1), primal and dual optimal function values are equal, ergo $f_{SM}(\overline{w}, \overline{b}, \overline{\xi}) = 0$ for all global solution $\left(\overline{w},\ \overline{b},\ \overline{\xi}\right)$ of the primal problem (P_{SM}). It is obvious that $f_{SM}(\overline{w}, \overline{b}, \overline{\xi}) = 0$ implies that $\overline{w} = 0$ and $\overline{\xi} = 0$, which is absurd, because using the constraints of (P_{SM}) we obtain that $\overline{b} \geq 1$ and $\overline{b} \leq -1$.

2. Analogous to the proof of Proposition 2.3.3.

3. Due to Remark 3.3.2, we can consider $(\overline{w},\ \overline{\eta},\ \overline{\nu})$ a solution for $(\widehat{D_{SM}})$, where $\overline{\nu}_i = C - \overline{\eta}_i$, for $i = 1, ..., m$. Due to Wolfe Duality (Theorem A.2.1) we have that primal and dual optimal function values are equal, ergo

$$f_{SM}(\overline{w},\ \overline{b},\ \overline{\xi}) - \hat{g}_{SM}(\overline{w},\ \overline{\eta},\ \overline{\nu}) = 0 \Longleftrightarrow \overline{w}^T\overline{w} + C\sum_{i=1}^{m} \overline{\xi}_i - \sum_{i=1}^{m} \overline{\eta}_i = 0.$$

Using in the previous formula the equalities $\overline{w} = \sum_{i=1}^{m} \overline{\eta}_i y_i x_i$ and $\sum_{i=1}^{m} y_i \overline{\eta}_i = 0$, it follows that

$$\sum_{i=1}^{m} \overline{\eta}_i y_i \overline{w}^T x_i + C \sum_{i=1}^{m} \overline{\xi}_i - \sum_{i=1}^{m} \overline{\eta}_i + b \sum_{i=1}^{m} y_i \overline{\eta}_i + \sum_{i=1}^{m} \overline{\eta}_i \overline{\xi}_i - \sum_{i=1}^{m} \overline{\eta}_i \overline{\xi}_i = 0,$$

which is the same as

$$\sum_{i=1}^{m} \overline{\eta}_i \left[y_i \left(\overline{w}^T x_i + b \right) - 1 + \overline{\xi}_i \right] + \sum_{i=1}^{m} \overline{\xi}_i \left(C - \overline{\eta}_i \right) = 0,$$

and, due to the admissibility of the primal and dual problem, we know that all terms are positive, therefore the result is concluded.

□

Remark 3.3.3 (Calculating the hyperplane after solving (D_{SM}))**.** *Let $\overline{\eta}$ be a global solution for (D_{SM}). Due to Proposition 3.3.2 we have that $\overline{w} = XY\overline{\eta}$.*

On the other hand, let us suppose that there is $i \in \{1, ..., m\}$ verifying that $0 < \overline{\eta}_i < C$ [1], by the second complementary condition we get that $\overline{\xi}_i = 0$, and using the first complementary condition it follows that

$$\overline{b} = y_i - \overline{w}^T x_i.$$

In order to avoid errors and to have a higher accuracy, as in [9], we propose to take the average of all possible values of $\overline{b}$, i.e. if $\Omega = \{i : 0 < \overline{\eta}_i < C\}$, then

$$\overline{b} = \frac{1}{\#\Omega} \left(\sum_{i \in \Omega} y_i - \overline{w}^T x_i \right),$$

where $\#\Omega$ means the cardinal of the set Ω.

In order to know the quality of the hyperplane we have obtained, we must calculate $\overline{\xi}$ to determine how many points it misclassifies, but instead of computing these values the classification function is usually used in practice. If we do not obtain good results, a useful option in practice is to vary the values of C as we did for (P_{SM}).

[1] If there is not $\overline{\eta}_i$ such that $0 < \overline{\eta}_i < C$, then we have a degenerate solution, see [11].

Chapter 4

Non linear SVM

There is a rather extensive theory about what is presented in this chapter, but we have decided to explain the basic theory that allows us to have some mathematical background of the concepts behind the software that we will use in Chapter 5. The main reference is [12].

4.1 Duality with Nonlinear SVM

In the previous chapter we have seen how to find hyperplanes that try to separate binary data sets. These hyperplanes may not be very accurate, so we wonder: Can we use a transformation function $\Phi : \mathbb{R}^n \to \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space, to map our data set

$$S = \{(x_i, y_i) : x_i \in \mathbb{R}^n, y_i \in \{-1, 1\}, i = 1, ..., m\},$$

to a new data set $\widehat{S} = \{(\Phi(x_i), y_i) : \Phi(x_i) \in \mathcal{H}, y_i \in \{-1, 1\}, i = 1, ..., m\}$, in order to obtain, maybe, a better decision boundary than the linear one (hyperplane) associated with the initial data set [1]? The answer is affirmative. Let us see an example.

Example 4.1.1. We consider in $\mathbb{R}^2$ the non linearly separable data points

$$x_1 = (0,\ 0)^T,\ x_2 = (3,\ 0)^T,\ x_3 = (0,\ 2)^T,\ x_4 = (-2,\ 0)^T \text{ and } x_5 = (0,\ -3)^T,$$

with the classification $x_1 \in S_{-1}$ and $x_2, x_3, x_4, x_5 \in S_1$, which represented in the left graph of Figure 4.1.

It is clear that the data can be separated by an ellipse, centered at the origin. So, let us consider a quadratic equation instead of a linear equation for separating the data of this example. For every point $x = (x_1,\ x_2)^T \in \mathbb{R}^2$, the quadratic equation

[1] In the sense that the new decision boundary is related to a lower number of misclassified points than the one formed by the hyperplane.

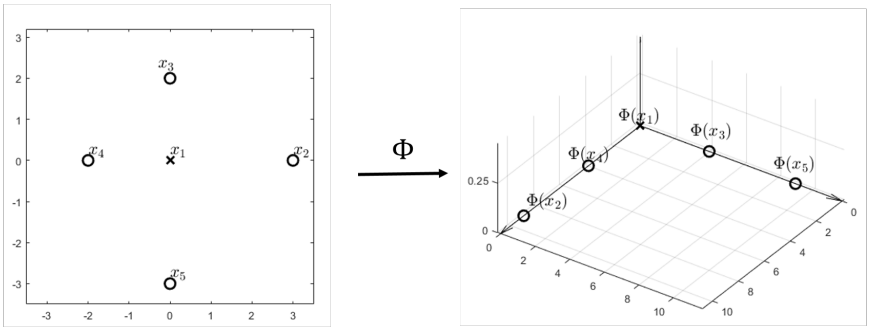

Figure 4.1: Graphs of Example 4.1.1. Class S_1 is represented by $\bigcirc$ and class S_{-1} by $\times$.

involves some linear combination of the terms x_1^2, x_2^2 and x_1x_2. So, we will work with an expression with the form $w^T\Phi(x) + b$ with the function $\Phi : \mathbb{R}^2 \to \mathbb{R}^3$, given by $\Phi(x) = (x_1^2,\ x_2^2,\ \sqrt{2}x_1x_2)^T$. If we apply Φ to each data point x_i, we obtain the new data set

$$\Phi(x_1) = (0,\ 0,\ 0)^T,\ \Phi(x_2) = (9,\ 0,\ 0)^T,$$

$$\Phi(x_3) = (0,\ 4,\ 0)^T,\ \Phi(x_4) = (4,\ 0,\ 0)^T \text{ and } \Phi(x_5) = (0,\ 9,\ 0)^T,$$

we will name it $\widehat{S}$, and it is shown in the right graph of Figure 4.1. As we can see, in this case the transformed data set is in $\mathbb{R}^3$ and linearly separable and , therefore, if we work with $\widehat{S}$ we will obtain a better classification function than the one obtained using S, because it will missclasified less points.

Now, let us make some considerations about the transformation function Φ.

1. Let us work with the objective function of the dual problem (D_{SM}) (see Proposition 3.3.1) for $\widehat{S}$. If $\widehat{X}$ is the matrix whose columns are $\Phi(x_i)$, with $x_i \in S_1 \cup S_{-1}$, then

$$g_{SM}(\eta) = \frac{1}{2}\eta^T Y \widehat{X}^T \widehat{X} Y \eta - e^T\eta = \frac{1}{2}\sum_{i,j=1}^{3} \eta_i\eta_j y_i y_j \Phi(x_i)^T\Phi(x_j) - e^T\eta.$$

2. Notice that $\widehat{X}^T\widehat{X}$ can be computed without using the explicit formula of Φ because

$$\Phi(x_i)^T\Phi(x_j) = (x_i^T x_j)^2.$$

So, we will be more efficient if we compute directly the values $(x_i^T x_j)^2$ (notice they are computed in $\mathbb{R}^2$ and the products $\Phi(x_i)^T\Phi(x_j)$ are computed in $\mathbb{R}^3$), without using the explicit formula of Φ.

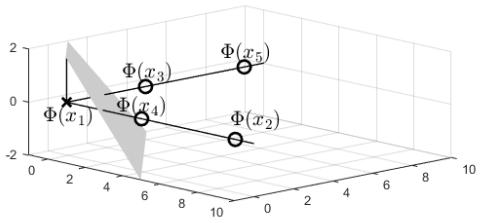

Figure 4.2: Plot of $\widehat{S}$ and the hyperplane from Example 4.1.1. Class S_1 is represented by $\bigcirc$ and class S_{-1} by $\times$.

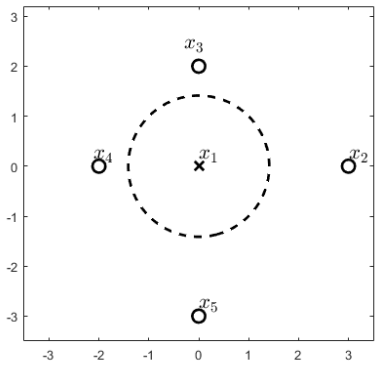

Figure 4.3: Plot of S and the decision boundary (dash line) from Example 4.1.1. Class S_1 is represented by $\bigcirc$ and class S_{-1} by $\times$.

Finally, notice that the coefficients of the hyperplane obtained after solving (D_{SM}) with $\widehat{S} \subset \mathbb{R}^3$ will be the coefficients of an ellipse that separates the original data in $\mathbb{R}^2$. Let us explain it. If we use the solver *quadprog* from Matlab to solve (D_{SM}) with $C = 1$ and $\widehat{S}$ and then we apply Remark 3.3.3, we get that $\overline{w} = (0.5,\, 0.5,\, 0)^T$ and $\overline{b} = -1$. In Figure 4.2 we can see the plot of the data set $\widehat{S}$ and the hyperplane whose coefficients are $\overline{w}$ and $\overline{b}$. If we consider the expression $\overline{w}^T\Phi(x) + \overline{b} = 0$ where $x \in \mathbb{R}^2$, we have that the decision boundary of S becomes $0.5x_1^2 + 0.5x_2^2 - 1 = 0$, for $x = (x_1,\, x_2)^T \in \mathbb{R}^2$, see Figure 4.3. So, we have got the nonlinear classification function defined by $class(x) = 0.5x_1^2 + 0.5x_2^2 - 1$ in $\mathbb{R}^2$.

Following the idea of the above procedure we can define other nonlinear classification functions, and that is why this method is named nonlinear SVM.

Let us formulate the optimization problem related to the transformation function, if we have a binary data set

$$S = \{(x_i, y_i) : x_i \in \mathbb{R}^n, y_i \in \{-1, 1\}, i = 1, ..., m\},$$

we are interested in dual optimization problems with the form

$$(D_\Phi) \begin{cases} min & g_\Phi(\eta) = \frac{1}{2}\eta^T Y \left(\Phi(X)^T\Phi(X)\right) Y\eta - e^T\eta \\ \text{subject to} & \eta \in \mathbb{R}^m, \\ & y^T\eta = 0, \\ & 0 \leq \eta \leq C. \end{cases}$$

where $\Phi : S_1 \cup S_{-1} \to \mathcal{H}$, is the function that transforms the data set into a new one, and $\Phi(X)$ is the matrix whose columns are $\Phi(x_i)$.

Remark 4.1.1. *Let us know that* (D_Φ) *is a convex quadratic optimization problem. Moreover, the results that have been seen in the previous chapter, for* (D_{SM})*, are also valid for* (D_Φ)*.*

Remark 4.1.2 (About the classification function)**.** *Let* $\overline{\eta}$ *be a global solution for* (D_Φ) *and* $SV = \{i : \overline{\eta}_i > 0\}$*, i.e. the set of indexes of the support vectors. Then*

$$\overline{w} = \Phi(X)Y\overline{\eta} = \sum_{i \in SV} \overline{\eta}_i y_i \Phi(x_i)$$

is the vector of the associated hyperplane and $\overline{b}$ *can be computed as*

$$\overline{b} = \frac{1}{\#\Omega}\left(\sum_{j\in\Omega} y_j - \overline{w}^T\Phi(x_j)\right) = \frac{1}{\#\Omega}\left(\sum_{j\in\Omega}\left(y_j - \sum_{i\in SV}\overline{\eta}_i y_i \Phi(x_i)^T\Phi(x_j)\right)\right),$$

where $\Omega = \{i : 0 < \overline{\eta}_i < C\}$*.*

Therefore, the classification function becomes

$$class(x) = sign\left(\sum_{i\in SV}\overline{\eta}_i y_i \Phi(x_i)^T\Phi(x) + \overline{b}\right).$$

Notice that neither in (D_Φ) nor in the classification function we have needed an explicit formula of the transformation function Φ. We have only required the values $\Phi(x_i)^T\Phi(x_j)$. That is why, in the following section we will present a type of functions that has a characterization that will be useful.

4.2 Kernel Functions

There are several definitions of kernel functions, all of them are equivalent. We opt to do something similar to what [12] does on page 30, although we have adapted their definition because they consider more general cases and sets that in principle do not need to be finite.

Definition 5 (**Gram Matrix**)**.** *Given a symmetric function* [2] $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ *and* $x_1,\ x_2,\ ..., x_r \in \mathbb{R}^n$*, then the* $r \times r$ *matrix*

$$G_K[x_1, ..., x_r] := \begin{pmatrix} K(x_1,x_1) & K(x_1,x_2) & \cdots & K(x_1,x_r) \\ K(x_2,x_1) & K(x_2,x_2) & \cdots & K(x_2,x_r) \\ \vdots & \vdots & \ddots & \vdots \\ K(x_r,x_1) & K(x_r,x_2) & \cdots & K(x_r,x_r) \end{pmatrix}$$

is called the Gram matrix of K *respect to* $x_1,\ x_2,\ ..., x_r$*.*

[2] A function $h : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is symmetric if it verifies $h(x, y) = h(y, x)$, $\forall x, y \in \mathbb{R}^n$.

Remark 4.2.1. *It is clear that $G_K[x_1, ..., x_r]$ is a symmetric matrix, because K is a symmetric function.*

Definition 6 (**(Positive Semidefinite) Kernel**). *Let be $\mathcal{X} = \{x_1, x_2, ..., x_m\} = S_1 \cup S_{-1}$. A symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ whose Gram matrix $G_K[x_1, ..., x_m]$ is a positive semidefinite matrix is called a positive semidefinite kernel (kernel for short).*

Now, we present the result that will be key to connect nonlinear SVM problems with kernels. That is to say the result that ensures us that behind every kernel there is a transformation function, [12].

Proposition 4.2.1 (Characterization of kernels). *Consider $\mathcal{X} = S_1 \cup S_{-1}$, and a symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Then K is a kernel if an only if there is a transformation function $\Phi : \mathcal{X} \to \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space, such that $K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$, for all $i, j \in \{1, 2, ..., m\}$.*

Proof. Fist, let us assume that K is a kernel. As K is a symmetric function over $x_1, ..., x_m$, it follows that the matrix $G_K[x_1, ..., x_m]$ is symmetric. As $G_K[x_1, ..., x_m]$ is a real symmetric matrix we have that $G_K[x_1, ..., x_m] = V \Sigma V^T$ where $\Sigma \in \mathbb{R}^m$ is the diagonal matrix whose diagonal values are the real eigenvalues of $G_K[x_1, ..., x_m]$ and $V \in \mathbb{R}^{m \times m}$ is an orthogonal matrix. Also, we know that the eigenvalues are nonnegative because $G_K[x_1, ..., x_m]$ is a positive semidefinite matrix.

To simplify the notation we consider $L = V\sqrt{\Sigma}$ [3], ergo $G_K[x_1, ..., x_m] = LL^T$. Then, if we consider the mapping $\Phi(x_i) = (L_{i,1}, L_{i,2}, ..., L_{i,m})^T \in \mathbb{R}^m$, it follows that

$$\Phi(x_i)^T \Phi(x_j) = (LL^T)_{i,j} = (V \Sigma V^T)_{i,j} = (G_K[x_1, ..., x_m])_{i,j} = K(x_i, x_j).$$

Now, let us assume that there is a mapping Φ such that $K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$. It is enough to check if the matrix $G_K[x_1, ..., x_m]$, whose elements are $K(x_i, x_j)$, is positive semidefinite, which is true because for all $u \in \mathbb{R}^m$ it is verified that

$$u^T G_K[x_1, ..., x_m] u = \sum_{i,j=1}^{m} u_i u_j K(x_i, x_j) = \sum_{i,j=1}^{m} u_i u_j \Phi(x_i)^T \Phi(x_j) = \left\| \sum_{i=1}^{m} u_i \Phi(x_i) \right\|_2^2 \geq 0.$$

□

Remark 4.2.2. *The previous proposition helps us to reformulate the optimization problem (D_Φ) in terms of kernels: instead of $\Phi(X)^T \Phi(X)$ we can use the matrix $G_K[x_1, ..., x_m]$. We are interested in kernel functions that can be computed efficiently*

[3] $\sqrt{\Sigma}$ means the diagonal matrix whose diagonal elements are the roots of the eigenvalues of $G_K[x_1, ..., x_m]$.

without using the transformation function Φ, which would imply working in a higher dimensional space than the original one.

Let us note that, using the notation from Remark 4.1.2, the classification function becomes

$$class(x) = sign\left(\sum_{i\in SV} \overline{\eta}_i y_i K(x_i, x) + \overline{b}\right),$$

where $\overline{b} = \dfrac{1}{\#\Omega}\left(\sum_{j\in\Omega}\left(y_j - \sum_{i\in SV}\overline{\eta}_i y_i K(x_i, x_j)\right)\right)$.

Therefore, we can avoid the use of the transformation function Φ for the resolution of (D_Φ) and the computation of the classification function, and we can see that K is the kernel the SVM method.

The process of incorporating kernels into soft margin SVM is commonly known as the kernel trick. This technique allows SVM to handle nonlinear relationships between data points and improve their classification capabilities.

Remark 4.2.3. *It is obvious that K is not uniquely defined in terms of one Φ. For example, if $x = (x_1,\ x_2)^T \in \mathbb{R}^2$, notice that $\Phi_1(x) = (x_1^2,\ x_2^2,\ \sqrt{2}x_1x_2)^T$ and $\Phi_2(x) = (x_1^2,\ \sqrt{2}x_1x_2,\ x_2^2)^T$ produce the same kernel. It can even happen the case where Φ's in different dimensions produce the same kernel, for example notice that $\Phi_3(x) = (x_1^2,\ x_2^2,\ x_1x_2,\ x_2x_1)$, leads to the same kernel as Φ_1 or Φ_2.*

The following proposition is useful, since it allows us to create admissible kernels from other kernels.

Proposition 4.2.2 (References [6] and [12])**.** *Given $a \in \mathbb{R}, a \geq 0$, the binary data set $S = \{(x_i, y_i) : x_i \in \mathbb{R}^n, y_i \in \{-1, 1\}, i = 1, ..., m\}$, $\mathcal{X} = S_1 \cup S_{-1}$, the sequence of kernels $\{K_t\}_{t\in\mathbb{N}}$, $K_t : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$, and the function $f : \mathbb{R}^n \to \mathbb{R}$, then the following expression also define kernels*

1. $K(x_i, x_j) = K_{t_1}(x_i, x_j) + K_{t_2}(x_i, x_j)$, *with* $t_1, t_2 \in \mathbb{N}$.
2. $K(x_i, x_j) = K_{t_1}(x_i, x_j)K_{t_2}(x_i, x_j)$, *with* $t_1, t_2 \in \mathbb{N}$.
3. $K(x_i, x_j) = aK_{t_1}(x_i, x_j)$.
4. *If* $\lim_{t\to\infty} K_t(x_i, x_j)$ *exits* $\forall x_i, x_j \in \mathcal{X}$, *then* $K(x_i, x_j) = \lim_{t\to\infty} K_t(x_i, x_j)$ *is a kernel.*
5. $K(x_i, x_j) = exp\left(K_{t_1}(x_i, x_j)\right)$.
6. $K(x_i, x_j) = f(x_i)f(x_j)$.

Proof. 1. It is obvious because the sum of positive semidefinite matrices is also positive semidefinite.

2. In order to simplify the notation let us define $A = G_{K_{t_1}}[x_1, ..., x_m]$ and $B = G_{K_{t_2}}[x_1, ..., x_m]$. Analogous to what happend in the proof of Proposition 4.2.1 we have that $A = V^A \Sigma^A (V^A)^T = \sum_{i=1}^{m} \lambda_i^A v_i^A (v_i^A)^T$, where λ_i^A are the eigenvalues of A (remember that they are non negative because they are the eigenvalues of positive semidefinte matrix) and v_i^A the columns of V^A and analogously $B = V^B \Sigma^B (V^B)^T = \sum_{i=1}^{m} \lambda_i^B v_i^B (v_i^B)^T$, notice that we use the hyperindexes to differentiate the decompositions of the matrices A and B.

   Let us recapitulate, we have to prove that the matrix whose elements are $K_{t_1}(x_i, x_j)K_{t_2}(x_i, x_j)$ is semidefinite positive. So, let us compute the Hadamard product[4] of the matrices A and B, that is $A \circ B = \sum_{i,j=1}^{m} \lambda_i^A \lambda_i^B (v_i^A \circ v_j^B)(v_i^A \circ v_j^B)^T$.

   Let us now prove that $A \circ B$ is a positive semidefinite matrix, so let be $u \in R^m$, then

$$u^T (A \circ B) u = \sum_{i,j=1}^{m} \lambda_i^A \lambda_i^B u^T (v_i^A \circ v_j^B)(v_i^A \circ v_j^B)^T u =$$

$$= \sum_{i,j=1}^{m} \lambda_i^A \lambda_i^B \left((v_i^A \circ v_j^B)^T u\right)^T \left((v_i^A \circ v_j^B)^T u\right) = \sum_{i,j=1}^{m} \lambda_i^A \lambda_i^B \left((v_i^A \circ v_j^B)^T u\right)^2 \geq 0.$$

3. It is obvious because the multiplication of a positive semidefinite matrix by a strictly positive number it is still a positive semidefinite matrix.

4. As K_t is a kernel for all $t \in \mathbb{N}$ we have that $u^T G_{K_t}[x_1, ..., x_m]u \geq 0, \forall u \in \mathbb{R}^n$. So, if $u \in \mathbb{R}^n$, we have that $0 \leq \lim_{t \to \infty} u^T G_{K_t}[x_1, ..., x_m]u = u^T G_K[x_1, ..., x_m]u$, where in the last equality we have used that fact of being pointwise limit.

5. It follows from the Taylor expansion of $exp\,(K_1(x_i, x_j))$ and the previous items of this proposition.

6. It follows from considering f as the mapping in Proposition 4.2.1.

□

There are many examples of kernels, but we will only mention the ones that we will use for our numerical results (Chapter 5).

[4]Given the matrices $P = (p_{ij}) \in \mathbb{R}^{s \times t}$ and $Q = (q_{ij}) \in \mathbb{R}^{s \times t}$, then the Hadamard product of P and Q is the matrix $P \circ Q = (p_{ij} q_{ij}) \in \mathbb{R}^{s \times t}$. It is also known as element-wise product.

- **Linear kernel** $K_L(x_i, x_j) := x_i^T x_j$.

  It is obvious that K_L is a kernel, just take Φ as the identity function.

  Notice that the linear kernel leads to the dual problem (D_{SM}).

- **Polynomial kernel** $K_P(x_i, x_j) := \left(x_i^T x_j + r\right)^d$, where d is a non-negative integer and $r \in \mathbb{R}, r \geq 0$.

  To justify that it is a kernel, it is enough to use Newton's Binomial Theorem and items (1), (2), (3) and (6) of Proposition 4.2.2.

- **Gaussian kernel** [5] $K_G(x_i, x_j) := exp\left(-\gamma \|x_i - x_j\|_2^2\right)$ where $\gamma > 0$.

  To justify that it is a kernel, notice that we can reformulate K_G as follows

  $$exp\left(-\gamma \|x_i\|_2^2\right) exp\left(-\gamma \|x_j\|_2^2\right) exp\left(2\gamma x_i^T x_j\right).$$

  The factor $exp\left(-\gamma \|x_i\|_2^2\right) exp\left(-\gamma \|x_j\|_2^2\right)$ is a kernel due to item (6) of Proposition 4.2.2 and using item (5) of the same proposition we have that $exp\left(2\gamma x_i^T x_j\right)$ is a kernel. So, using item (2) of Proposition 4.2.2 the result follows.

The values d, r, γ are known as the kernel parameters.

[5] It is also named as RBF kernel, where RBF means Radial Basis Function.

Chapter 5

Numerical practice

The objective of this chapter is to experiment with some software that internally uses what was described in Chapters 3 and 4. We will use Scikit-learn (shortly Sklearn), see [10]. It is a free software machine learning library for the programming language Python. Although Sklearn is mostly written in Python, it incorporates the C++ libraries LIBSVM (see [4]) and LIBLINEAR (see [8]) that provide reference implementations of SVM.

We will use the Breast Cancer Wisconsin (Diagnostic) Data Set. It was created by Dr. William H. Wolberg, W. Nick Street and Olvi L. Mangasarian in 1995, from the University of Wisconsin. The data were collected in order to perform a study in which the type of breast tumor (benign or malignant) could be diagnosed using different techniques depending on a series of characteristics extracted from some cell nucleus.

We want to compute some classifier models that helps us to decide to which class a new tumor will belong to, i.e. if it is a benign or malignant tumor. We will use a technique widely used in Machine Learning. First, we will divide the data set into the groups: train_data (70% of the data set) and test_data (30% of the data set). Then, we will compute the SVM model using the train_data. And finally, in order to evaluate the model, we will predict the membership class of each tumor from the test_data, and then compare it with the actual belonging class.

In order to compare the models, all of them will be computed using the same partition of the data set.

5.1 Analyzing Classification Models

In this section we will consider methods provided in the library Sklearn that we will use to evaluate the quality of the models.

Confusion Matrix

After predicting the membership class of the test_data, we will construct the confusion matrix, which is represented in Table 5.1. As we can see in it, the actual classification values are compared with those predicted by the model. Therefore, we want large values in the main diagonal and small values in the antidiagonal to get good models.

	Predicted Negative	**Predicted Positive**
Actual Negative	True Negative (TN)	False Positive (FP)
Actual Positive	False Negative (FN)	True Positive (TP)

Table 5.1: Confusion matrix.

Metrics

From the values obtained in the confusion matrix, we will calculate the following metrics:

- The proportion between the correct predictions and total of predictions:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}.$$

- The proportion between the true positive samples and the actual positive samples

$$Recall = \frac{TP}{TP + FN}.$$

- The proportion between the true positive samples and the predicted positive samples

$$Precision = \frac{TP}{TP + FP}.$$

  If the division is not well defined, we will will take 0 as value.

We will use accuracy as the principal metric. Also, we will measure the time needed to construct and evaluate the models.

Feature	**Minimum**	**Maximum**
Radius (mean)	6.981	28.11
Texture (mean)	9.71	39.28
Perimeter (mean)	43.79	188.5
Area (mean)	143.5	2501.0
Smoothness (mean)	0.053	0.163
Compactness (mean)	0.019	0.345
Concavity (mean)	0.0	0.427
Concave points (mean)	0.0	0.201
Symmetry (mean)	0.106	0.304
Fractal dimension (mean)	0.05	0.097
Radius (standard error)	0.112	2.873
Texture (standard error)	0.36	4.885
Perimeter (standard error)	0.757	21.98
Area (standard error)	6.802	542.2
Smoothness (standard error)	0.002	0.031
Compactness (standard error)	0.002	0.135
Concavity (standard error)	0.0	0.396
Concave points (standard error)	0.0	0.053
Symmetry (standard error)	0.008	0.079
Fractal dimension (standard error)	0.001	0.03
Radius (worst)	7.93	36.04
Texture (worst)	12.02	49.54
Perimeter (worst)	50.41	251.2
Area (worst)	185.2	4254.0
Smoothness (worst)	0.071	0.223
Compactness (worst)	0.027	1.058
Concavity (worst)	0.0	1.252
Concave points (worst)	0.0	0.291
Symmetry (worst)	0.156	0.664
Fractal dimension (worst)	0.055	0.208

Table 5.2: Data set features and the minimum and maximum value of each feature. Mean represents the mean of the values measured for the cells nucleus of the image. Worst means the mean of the three worst/largest values of the cells nucleus of the image.

5.2 Breast Cancer Wisconsin (Diagnostic) Data Set

5.2.1 About the Data Set

The Breast Cancer Wisconsin (Diagnostic) Data Set has 569 samples (212 malignant tumors and 357 benign tumors) with 30 features each. We have taken it from [7]. As we have said before, we will divide the data set into two groups. In Table 5.3 we can see the partition of the data set [1].

	train_data	test_data
Number of benign tumors	252	105
Number of malignant tumors	146	66

Table 5.3: Size of the train_data and test_data.

Let us see, as a general idea, how the data set is formed. Given a digitized image of a fine needle aspirate (FNA) of a breast mass, then for each cell nucleus was measured the following attributes:

- Radius (mean of distances from center to points on the perimeter).
- Texture (standard deviation of gray-scale values).
- Perimeter.
- Area.
- Smoothness (local variation in radius lengths).
- Compactness (perimeter^2 / area - 1.0).
- Concavity (severity of concave portions of the contour).
- Concave points (number of concave portions of the contour).
- Symmetry.
- Fractal dimension ("coastline approximation" - 1).

[1] To do the partition we use the Sklearn´s function `sklearn.model_selection.train_test_split`. In order to always obtain the same results, we have set the random seed as 101, although we have obtained similar models if we vary the seed.

From the attributes measured for some cell nucleus, of each breast mass, there are computed the features shown in Table 5.2. The second and third columns of Table 5.2 represents the minimum and maximum value of each feature of the data set. Notice that in some features there is a substantial difference between the minimum and maximum value, so scaling the data will be something to consider when using the numerical computation.

5.2.2 Numerical Results

For the numerical experiments we have used the function `sklearn.svm.SVC`, which internally solves the dual problem (D_Φ), see Section 4.1.

Linear Kernel

Remember that the linear kernel is defined by

$$K_L(x_i, x_j) = x_i^T x_j.$$

As we can see it does not depend on any parameters, ergo this type of kernel does not give us much room for maneuver. In Table 5.4 we can see the confusion matrix and metrics of the linear kernel model for some values of the upper bound C and the original data (without any manipulation).

	Confusion matrix				**Metrics**			
	Actual benign		**Actual malignant**					
	Predicted benign (TN)	**Predicted malignant (FP)**	**Predicted benign (FN)**	**Predicted malignant (TP)**	**Acuracy**	**Recall**	**Precision**	**Time**
$C = 0.1$	101	4	8	58	0.9298	0.8788	0.9355	0.6s
$C = 1$	102	3	7	59	0.9415	0.8939	0.9516	2.4s
$C = 10$	102	3	6	60	0.9474	0.9091	0.9524	3.2s
$C = 10^2$	104	1	8	58	0.9474	0.8788	0.9831	9.7s
$C = 10^3$	102	3	7	59	0.9415	0.8939	0.9516	4.5s

Table 5.4: Confusion matrix and metrics: linear kernel with non standardized data.

On the other hand, if we standardize[2] the data set before doing the partition, we obtain the results represented in Table 5.5. As we can see, there are significant differences between the computation times if we standardize the data or not. In addition, by standardizing the data we have reduced the number of false negatives, i.e. the actual malignant tumors that the model classifies as benign.

[2]Remember that standardizing a sample $z = \{z_i\}_{i=1}^p \subset \mathbb{R}, p \in \mathbb{N}$, means to compute the values $\left\{\frac{z_i - \overline{z}}{var(z)}\right\}_{i=1}^p \subset \mathbb{R}$ if $var(z) \neq 0$, where $var(z)$ means the variance of the sample and $\overline{z}$ the mean. To do it, we use the Sklearn's function `sklearn.preprocessing.StandardScaler`, which standardizes each feature.

	Confusion matrix				Metrics			
	Actual benign		Actual malignant					
	Predicted benign (TN)	Predicted malignant (FP)	Predicted benign (FN)	Predicted malignant (TP)	Acuracy	Recall	Precision	Time
$C = 0.1$	105	0	4	62	0.9766	0.9394	1.0000	0.1s
$C = 1$	103	2	2	64	0.9766	0.9697	0.9697	0.1s
$C = 10$	105	0	5	61	0.9708	0.9242	1.0000	0.1s
$C = 10^2$	102	3	4	62	0.9591	0.9394	0.9538	0.1s
$C = 10^3$	102	3	4	62	0.9591	0.9394	0.9538	0.1s

Table 5.5: Confusion matrix and metrics: linear kernel varing C.

As we have just seen, standardizing the data has improved the quality of our models. If we consider the computational time, we see that they are reduced by at least 5 times, but in other cases we have experimented with it has meant going from several minutes to tenths of seconds.

From now on, we will work with the standardized the data set.

Polynomial Kernel

Remember that the formula of the polynomial kernel is

$$K_P(x_i, x_j) = \left(x_i^T x_j + r\right)^d,$$

where d is a non-negative integer and $r \in \mathbb{R}, r \geq 0$. Notice that in this case we can vary the kernel parameters, r and d, and the regularization term, C.

Let us take $C = 0.1$. First of all let us have a look at the influence that the parameter d can have, therefore we will consider $r = 0$. The results for some values of d and the standardized data are represented in Table 5.6 [3].

	Confusion matrix				Metrics			
	Actual benign		Actual malignant					
	Predicted benign (TN)	Predicted malignant (FP)	Predicted benign (FN)	Predicted malignant (TP)	Acuracy	Recall	Precision	Time
$d = 1$	105	0	4	62	0.9766	0.9394	1.0000	0.1s
$d = 2$	101	4	17	49	0.8772	0.7424	0.9245	0.1s
$d = 3$	101	4	5	61	0.9474	0.9242	0.9385	0.1s
$d = 4$	91	14	21	45	0.7953	0.6818	0.7627	0.1s

Table 5.6: Confusion matrix and metrics: polynomial kernel with $C = 0.1$, $r = 0$ and varying d.

Since the best model we have in Table 5.6 is the one generated with $d = 1$, the natural thing now would be to see what happens if we keep $d = 1$ and vary r.

[3] It is clear that linear kernel is a particular case of polynomial kernel, ergo the first row of Table 5.6 is the same as the first row of Table 5.5. But, we have decided to include this row again so that the results on the influence of d become easier to compare at a look.

However, for $r \in \{0.01, 0.01, 0.5, 1, 5, 10\}$ we have obtained models with the same metrics.

Let us now see what happens if we take the worst model shown in Table 5.6, i.e. $d = 4$, and we try to vary r. As we can see in Table 5.7, all the models are better than the one with $r = 0$, and the best classifier model is the one that arises from taking $r = 5$.

	Confusion matrix				**Metrics**			
	Actual benign		**Actual malignant**					
	Predicted benign (TN)	**Predicted malignant (FP)**	**Predicted benign (FN)**	**Predicted malignant (TP)**	**Acuracy**	**Recall**	**Precision**	**Time**
$r = 0.1$	94	11	15	51	0.8480	0.7727	0.8226	0.1
$r = 0.5$	98	7	11	55	0.8947	0.8333	0.8871	0.1
$r = 1$	96	9	8	58	0.9006	0.8788	0.8657	0.1s
$r = 5$	101	4	3	63	0.9591	0.9545	0.9403	0.1s

Table 5.7: Confusion matrix and metrics: polynomial kernel with $C = 0.1$, $d = 4$ and varying r.

Now, let us take $C = 1$. As we have done before, let us see the influence of d while $r = 0$. In Table 5.8 are represented the results for some vales of d. As we can see, the best model arises from taking $d = 1$. Moreover, in Table 5.8 are contained the values obtained if we use the default parameters of the algorithm for the polynomial kernel, which are $C = 1$, $r = 0$ and $d = 3$.

	Confusion matrix				**Metrics**			
	Actual benign		**Actual malignant**					
	Predicted benign (TN)	**Predicted malignant (FP)**	**Predicted benign (FN)**	**Predicted malignant (TP)**	**Acuracy**	**Recall**	**Precision**	**Time**
$d = 1$	103	2	2	64	0.9766	0.9697	0.9697	0.1s
$d = 2$	90	15	19	47	0.8012	0.7121	0.7581	0.1s
$d = 3$	101	4	5	61	0.9474	0.9242	0.9385	0.1s
$d = 4$	91	14	21	45	0.7953	0.6818	0.7627	0.1s
$d = 5$	101	4	12	54	0.9064	0.8182	0.9310	0.1s

Table 5.8: Confusion matrix and metrics: polynomial kernel with $C = 1$, $r = 0$ and varying d.

As a general comment we can say that the computational time required for the different values of the polynomial kernel parameters and the regularization term are negligible. Regarding the best models among all those presented in this part:

- One is formed by by taking $C = 0.1$, $r = 0$ and $d = 1$ (first row of Table 5.6).
- And the other, it is generated by $C = 1$, $r = 0$ and $d = 1$ (first row of Table 5.8).

The difference between the models is that the first model predicts 100% of benign tumors and 93.94% of malignant tumors, and the second model loses quality in the detection of benign tumors (96.97%) but gains quality in the prediction of malignant tumors (96.97%).

Gaussian Kernel

Remember that the Gaussian kernel is

$$K_G(x_i, x_j) = exp\left(-\gamma \|x_i - x_j\|_2^2\right), \text{ where } \gamma > 0.$$

Let us start taking γ as the value used by the algorithm by default if we choose the Gaussian kernel in Sklearn, that value is $\gamma_d := \frac{1}{\#(features) \cdot var(all\ data)} \approx 0.0328$, i.e. it is inversely proportional to the product of the number of features and the variance of the entire data set. As we can see in Table 5.9 with γ_d the best model arises from taking $C = 1$, which are also the default parameters chosen for this kernel in Sklearn.

	Confusion matrix				**Metrics**			
	Actual benign		**Actual malignant**					
	Predicted benign (TN)	**Predicted malignant (FP)**	**Predicted benign (FN)**	**Predicted malignant (TP)**	**Acuracy**	**Recall**	**Precision**	**Time**
$C = 0.1$	103	2	10	56	0.9298	0.8485	0.9655	0.1s
$C = 1$	104	1	3	63	0.9766	0.9545	0.9844	0.1s
$C = 10$	103	2	3	63	0.9708	0.9545	0.9692	0.1s
$C = 10^2$	100	5	6	60	0.9357	0.9091	0.9231	0.1s
$C = 10^3$	100	5	6	60	0.9357	0.9091	0.9231	0.1s

Table 5.9: Confusion matrix and metrics: Gaussian kernel with $\gamma = \gamma_d$ and varying C.

Now, let us take $C = 0.1$, and let us try some values for γ. As we can see in Table 5.10 we require low values of γ to obtain good models. Moreover, notice that the models with $\gamma = 0.2$ and $\gamma = 0.3$ obtain numerous false negatives, so they are not effective models.

	Confusion matrix				**Metrics**			
	Actual benign		**Actual malignant**					
	Predicted benign (TN)	**Predicted malignant (FP)**	**Predicted benign (FN)**	**Predicted malignant (TP)**	**Acuracy**	**Recall**	**Precision**	**Time**
$\gamma = 0.01$	104	1	11	55	0.9298	0.8333	0.9821	0.1s
$\gamma = 0.1$	101	4	8	58	0.9298	0.8788	0.9355	0.1s
$\gamma = 0.2$	105	0	62	4	0.6374	0.0606	1.0000	0.1s
$\gamma = 0.3$	105	0	66	0	0.6140	0.0000	0.0000	0.1s

Table 5.10: Confusion matrix and metrics: Gaussian kernel with $C = 0.1$ and varying γ.

Let us now examine what happens if we take the regularization term greater than the previous one, for example $C = 1$. As we can see in Table 5.11 we need small values of γ for the models to have satisfactory quality, as before.

It is clear that the best model using the Gaussian kernel, of those presented previously, arises from taking $C = 0.1$ and $\gamma = \gamma_d$.

	Confusion matrix				Metrics			
	Actual benign		Actual malignant					
	Predicted benign (TN)	Predicted malignant (FP)	Predicted benign (FN)	Predicted malignant (TP)	Acuracy	Recall	Precision	Time
$\gamma = 0.01$	104	1	8	58	0.9532	0.9697	0.9143	0.1s
$\gamma = 0.1$	99	6	2	64	0.9532	0.9697	0.9143	0.1s
$\gamma = 0.2$ and $\gamma = 3$	97	8	3	63	0.9357	0.9545	0.8873	0.1s
$\gamma = 0.4$	103	2	22	44	0.8596	0.6667	0.9565	0.1s

Table 5.11: Confusion matrix and metrics: Gaussian kernel with $C = 1$ and varying γ.

Sklearn Selection of Parameter Values and Conclusions

It is obvious that we cannot check by hand a large number of values for the kernel parameters and the regularization parameter. In practice it is usually used the Sklearn function `sklearn.model_selection.GridSearchCV` (`GridSearchCV` shortly), which using multiple values for the kernel parameters and C it performs a grid [4] and, with the help of a 5-fold cross-validation [5], it looks for the best values from the supplied ones. The problem with this function is that it requires a high computation time, which is obviously influenced by the number of grid points and the size of the data set.

For the linear kernel, we are going to look for the best model with the penalization term C taking one of the values of the set $\{0.1, 1, 2, ..., 1000\}$, ergo we have 1001 candidates for C and, as with each candidate five models are fitted using 5-fold cross-validation, we have to compute 5005 models. After 63 seconds, we obtain that the best model arises from taking $C = 0.1$. It is redundant to write the confusion matrix and metrics for this model since it is already included in Table 5.5. Let us remember that we had previously obtained, see 5.5, the same accuracy value for the linear kernel with $C = 0.1$ as with $C = 1$. At that time we did not say which was the best, but `GridSearchCV` has helped us to decide that if we use a 5-fold cross validation it is better to use $C = 0.1$.

Regarding the polynomial kernel, we want to find the model with the best metrics verifying $d \in \{1, 2, ..., 5\}$ and $r \in \{0, 0.2, 0.4, ..., 9.8, 10\}$, and penalization term $C \in \{1, 5, 10, ..., 995, 1000\}$, which results in 40200 combinations and therefore 201000 models to compute and compare. After 67 minutes and 53 seconds, we obtain that the combination $d = 1, r = 0$ and $C = 1$ is the best option, and it is redundant to write the confusion matrix and metrics for this model since it is already included in Table 5.8.

Finally, for the Gaussian kernel, we are going to look for the best model that has

[4] That is to say that it makes all possible combinations of the different supplied values.

[5] This means that it divides the train_data into 5 groups. From these groups, it computes 5 models, using in each of them 4 of the groups and evaluating it using the remaining one. It does this process for each combination of the grid.

as the kernel parameter $\gamma \in \{\gamma_d, 0.01, 0.06, ..., 0.96\}$ and the penalization term $C \in \{1, 2, ..., 1000\}$, wich results in 20979 combinations and therefore 104895 models. After 35 minutes, we obtain that the best model rises from taking $\gamma = 0.01$ and $C = 3$. In Table 5.12 are represented the confusion matrix and the metrics of this model.

	Predicted benign	**Predicted malignant**
Actual benign	105	0
Actual malignant	3	63

Metrics	
Accuracy	0.9825
Recall	0.9545
Precision	1.0000

Table 5.12: Confusion matrix and metrics: Gaussian kernel with $\gamma = 0.01$ and $C = 3$.

Let us make the final conclusions about our models. In this section we have obtained classifier models with very good accuracies for different kernels, and in some cases with similar results. Taking into account the accuracy, among all the models the slightly better one is the one generated by the Gaussian kernel with $\gamma = 0.01$ and $C = 3$.

Model	Accuracy	Precission
Lineal kernel with $C = 0.1$	0.9766	1
Polynomial kernel with $C = 1, d = 1$ and $r = 0$	0.9766	0.9697
Gaussian kernel with $C = 1$ and $\gamma = \gamma_d$	0.9766	0.9844
Gaussian kernel with $C = 3$ and $\gamma = 0.01$	0.9825	1
Range of metrics in [7] using SVM	0.9021-0.9790	0.9041-0.9818

Table 5.13: Baseline Model Performance

Finally, in [7] we can see a range of values obtained for accuracy and precision using SVM. They are shown in the last row of Table 5.13. Moreover in this table we present our best values of accuracy and precision obtained. As we can see, the model that arises from taking the Gaussian kernel with $C = 3$ and $\gamma = 0.01$ has slightly better metrics than the ones presented in [7].

5.2.3 Decision Boundaries. Some Examples.

The objective of this section is to see how the decision boundary varies according to the kernel we use and, we are not going to worry about finding the best kernel parameters and regularization parameter. In order to plot the results, we will only use the first two features of the data set to compute the models, i.e. "Radius (mean)" and "Texture (mean)". We will also use the standardized data and the same partition of the data set.

For the plots of this section we will use the following scheme: Empty circles represent malignant tumors and filled circles benign tumors. The exterior circles represent the points of the test_data. The gray part corresponds to the area of malignant tumors [6], the white part corresponds to area of benign tumors, and the dash lines represents the boundary hyperplanes.

In Figure 5.1 we can see the plot for the linear kernel with $C = 1$, and in Figure 5.2 the linear kernel for $C = 1$.As we can see, the main difference between both figures is the separation margin.

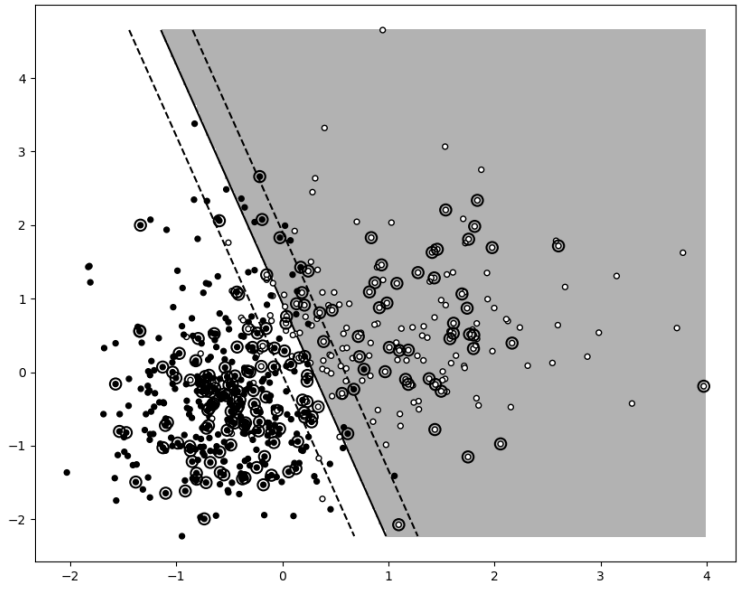

Figure 5.1: Decision boundary: linear kernel with $C = 0.1$.

In Figure 5.3 we can see the representation of the polynomial kernel for $d = 3$, $r = 0$ and $C = 0.1$, and in Figure 5.4 the polynomial kernel for $d = 3$, $r = 0$ and $C = 1$. As we can see, the main difference of both figures is in the curvature of the decision boundary in the central cloud of points.

Finally, in Figure 5.5 we can see the representation of the Gaussian kernel for $\gamma = 1$ and $C = 0.1$, and in Figure 5.6 the Gaussian kernel for $\gamma = 1$ and $C = 0.1$. As we can see there is a significant difference in the size and shape of the two areas between the figures.

As we have just seen, we can have several different decision boundaries, depending on what kernel we choose.

[6]This means the space in which a point will be classified as malignant tumor.

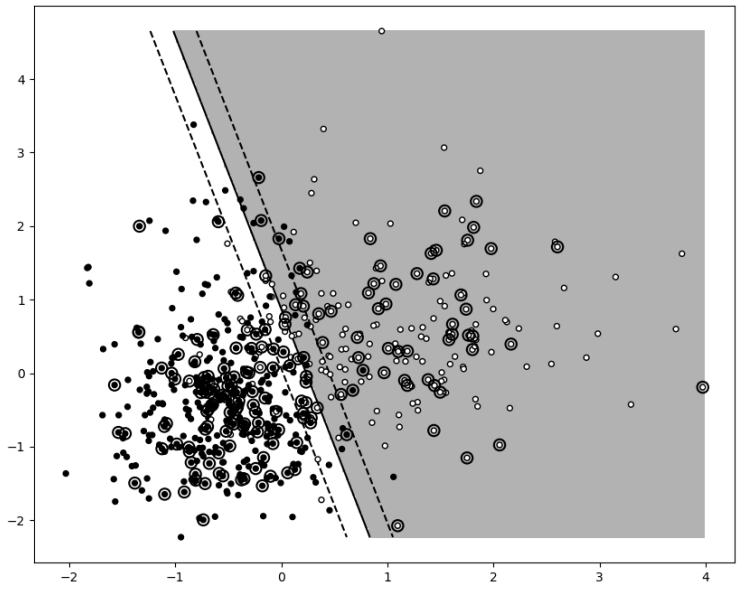

Figure 5.2: Decision boundary: linear kernel with $C = 1$.

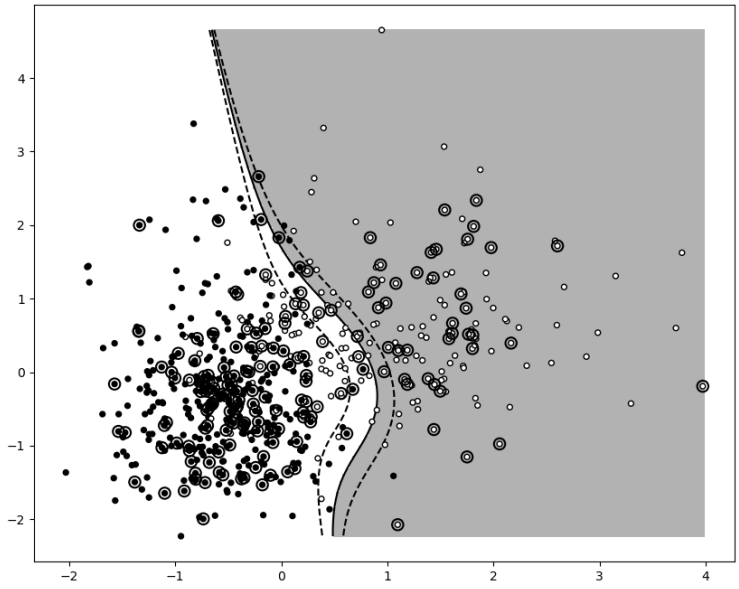

Figure 5.3: Decision boundary: polynomial kernel with $d = 3$, $r = 0$ and $C = 0.1$.

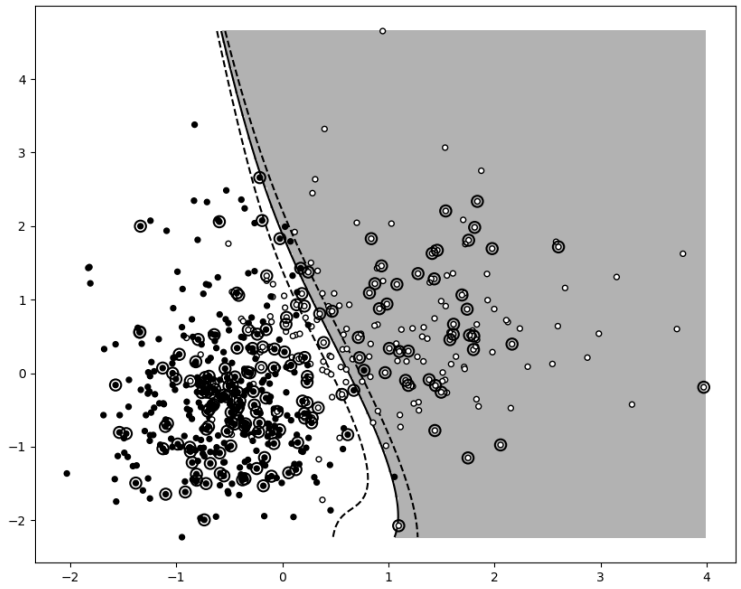

Figure 5.4: Decision boundary: polynomial kernel with $d = 3$, $r = 0$ and $C = 1$.

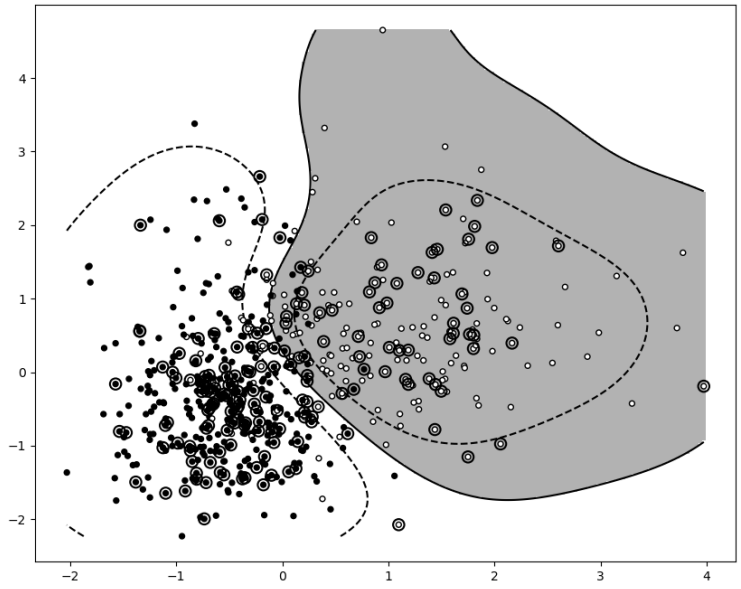

Figure 5.5: Decision boundary: Gaussian kernel with $\gamma = 1$ and $C = 0.1$.

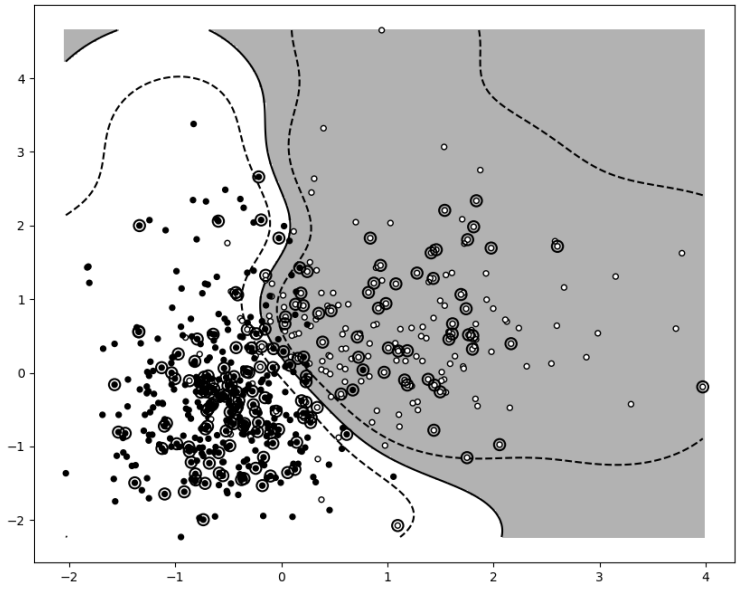

Figure 5.6: Decision boundary: Gaussian kernel with $\gamma = 1$ and $C = 1$.

Appendix A

Optimization theory

This appendix is written in order to introduce some concepts that will be necessary in the optimization theory used in this work, for more details and proofs see [2].

A.1 Basic concepts and results

Theorem A.1.1. *Given the open set $\Omega \subset \mathbb{R}^n$, the C^2 function $f : \Omega \longrightarrow \mathbb{R}$ and the convex set $K \subset \Omega$, then f is a convex function over K if and only if*

$$(y-x)^T \nabla^2 f(x)(y-x) \geq 0, \ \forall x, y \in K.$$

Remark A.1.1. *For function $f(x) = \frac{1}{2}x^T Hx + p^T x$, the previous theorem states that the it is convex if the matrix H is positive or positive semidefinite.*

Definition 7 (**Coercive function**). *Given $f : K \subset \mathbb{R}^n \longrightarrow \mathbb{R}$, then it is said that f is a coercive function if it verifies*

$$f(x_n) \xrightarrow{n \to +\infty} +\infty, \ \forall \ \{x_n\}_{n \in \mathbb{N}} \subset K \text{ verifying } \|x_n\| \xrightarrow{n \to +\infty} +\infty.$$

From now on, to write the following results of this section we will consider the following general optimization problem

$$\begin{cases} min & f(x) \\ \\ \text{subject to} & x \in K, \end{cases} \tag{A.1}$$

where $K \subset \mathbb{R}^n, K \neq \emptyset$, and $f : K \longrightarrow \mathbb{R}$.

Definition 8 (**Feasible point**). *Consider problem* (A.1)*, if $x \in \mathbb{R}^n$ verifies that $x \in K$, then it is said that x is a feasible point.*

Definition 9 (**Feasible set**). *Consider problem* (A.1)*, the set of all feasible points of the problem is known as feasible set.*

Definition 10 (**Convex optimization problem**). *Problem* (A.1) *is known as a convex optimization problem if K is a convex set and $f \in C^1$ is a convex function over K.*

Definition 11 (**Quadratic optimization problem**). *Problem* (A.1) *is known as quadratic optimization problem if $f(x) = \frac{1}{2}x^T Hx + p^T x$, where $H \in \mathbb{R}^{n\times n}$ is a symmetric matrix, $p \in \mathbb{R}^n$ and K is a set of linear constraints.*

Theorem A.1.2. *Given problem* (A.1)*, if K is a closed set and f is a coercive continuous function over K, then there is at least one global solution for problem* (A.1).

Theorem A.1.3. *Given problem* (A.1)*, if K is a convex set and f is convex function over K (i.e. the problem is convex), then each local solution for the problem it is also a global solution.*

A.1.1 Lagrange Multipliers

In this section we will consider the following optimization problem

$$\begin{cases} min & f(x) \\ \\ \text{subject to} & x \in \Omega \subset \mathbb{R}^n, \\ & h_i(x) = 0, \quad \text{for } i = 1, ..., n_I, \\ & g_j(x) \leq 0, \quad \text{for } j = 1, ..., n_D, \end{cases} \tag{A.2}$$

where $f, h_i, g_j : \Omega \longrightarrow \mathbb{R}$, and $\Omega \subset \mathbb{R}^n$ is an open subset.

Theorem A.1.4 (Lagrange multipliers rule). *Let $f, h_i, g_j : \Omega \longrightarrow \mathbb{R}$ be C^1 functions, for $i = 1, ..., n_I$ and for $i = 1, ..., n_D$. If there is $\overline{x} \in \Omega$ solution for problem* (A.2)*, then there are $1 + n_I + n_D$ numbers that: $\overline{\alpha} \in \mathbb{R}_+$, $\{\overline{\lambda}_i\}_{i=1}^{n_I} \subset \mathbb{R}$ and $\{\overline{\mu}_j\}_{j=1}^{n_D} \subset \mathbb{R}_+$ verifying:*

$$\overline{\alpha} + \sum_{i=1}^{n_I} |\overline{\lambda}_i| + \sum_{j=1}^{n_D} \overline{\mu}_j > 0, \tag{A.3}$$

$$\overline{\alpha}\nabla f(\overline{x}) + \sum_{i=1}^{n_I} \overline{\lambda}_i \nabla h_i(\overline{x}) + \sum_{j=1}^{n_D} \overline{\mu}_j \nabla g_j(\overline{x}) = 0, \tag{A.4}$$

$$\overline{\mu}_j \geq 0 \qquad and \qquad \overline{\mu}_j g_j(\overline{x}) = 0, \qquad for\ i = 1, ..., n_D, \tag{A.5}$$

$$h_i(\overline{x}) = 0, \quad for\ i = 1, ..., n_I \quad and \quad g_j(\overline{x}) \geq 0, \quad for\ i = 1, ..., n_D. \tag{A.6}$$

Corollary A.1.1. $\overline{\alpha} = 1$ *can be taken in Theorem A.1.4 if the constraints of problem* (A.2) *are linear.*

The inequality constraints verifying $g(\overline{x}) = 0$ are known as **active constraints** at $\overline{x}$. The four conditions (A.3)-(A.6) are named as Fritz John conditions. But in the case that $\overline{\alpha} = 1$ are known as **Kuhn-Tucker conditions**. Also if $\overline{x}$ verifies the Kuhn-Tucker conditions it is said that $\overline{x}$ is a **Kuhn-Tucker point**.

The numbers $\{\overline{\lambda}_i\}_{i=1}^{n_I}$ and $\{\overline{\mu}_j\}_{j=1}^{n_D}$ are known as **Lagrange multipliers**. Also, the condition on the right of (A.5) is known as **complementary condition**, and conditions (A.6) as **feasibility conditions**.

Theorem A.1.5 (First-order sufficient conditions)**.** *Consider problem* (A.2)*, where* $f, h_i, g_j : \Omega \longrightarrow \mathbb{R}$ *are* C^1 *functions, for* $i = 1, ..., n_I$ *and for* $i = 1, ..., n_D$*, such that:*

- *The feasible set is a convex set, i.e.*

$$K = \{x \in \Omega : h_i(x) = 0 \text{ for } i = 1, ..., n_I \text{ and } g_j(x) \leq 0 \text{ for } j = 1, ..., n_D\}$$

  is a convex set.

- f *is a convex function over* K*.*

If $\overline{x}$ *is a Kuhn-Tucker point, then* $\overline{x}$ *is a global solution for problem* (A.2)*.*

A.2 Wolfe Duality

The objective of this section is to obtain an alternative formulation to a given optimization problem. The original problem is known as **primal problem** and the other as **dual problem**.

Theorem A.2.1 (Wolfe Duality)**.** *Let* $\overline{x}$ *be a global solution for the following optimization problem*

$$\begin{cases} \text{min} & f(x) \\ \\ \text{subject to} & x \in \mathbb{R}^n, \\ & a_i^T x = b_i\,, \quad \text{for } i = 1, ..., n_I, \\ & g_j(x) \leq 0\,, \ \text{for } j = 1, ..., n_D, \end{cases}$$

where $f, g_i : \mathbb{R}^n \longrightarrow \mathbb{R}$ *are* C^1 *and convex functions. If* $(\overline{\lambda}, \overline{\mu})$ *are Lagrange multipliers associated with* $\overline{x}$*, then* $(\overline{x}, \overline{\lambda}, \overline{\mu})$ *is a global solution for the problem*

$$\begin{cases} min & L(x,\lambda,\mu) = f(x) + \sum_{i=1}^{n_I} \lambda_i (a_i^T x - b_i) + \sum_{j=1}^{n_D} \mu_j g_j(x) \\ \\ subject\ to & (x,\lambda,\mu) \in \mathbb{R}^n \times \mathbb{R}^{n_I} \times \mathbb{R}^{n_D}, \\ & \nabla_x L(x,\lambda,\mu) = 0, \\ & \mu_j \geq 0\,, \quad for\ j = 1, ..., n_D. \end{cases}$$

Also, it is verify that $L(\overline{x}, \overline{\lambda}, \overline{\mu}) = f(\overline{x})$.

The function $L(x, \lambda, \mu)$ is named **Lagrangian function**.

Bibliography

[1] S. Abe. Analysis of Support Vector Machines. In *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*, pages 89–98, 2002.

[2] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2016.

[3] C. J. C. Burges and D. Crisp. Uniqueness of the SVM Solution. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 223–229. MIT Press, 1999.

[4] C.-C. Chang and C.-J. Lin. LIBSVM: A Library for Support Vector Machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[5] C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning*, 20:273–297, 1995.

[6] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[7] D. Dua and C. Graff. UCI Machine Learning Repository. `http://archive.ics.uci.edu/ml`, 2017. University of California, Irvine, School of Information and Computer Sciences.

[8] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[9] I. Griva, S. G. Nash, and A. Sofer. *Linear and Nonlinear Optimization*. SIAM, 2009.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn:

Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[11] R. Rifkin, M. Pontil, and A. Verri. A Note on Support Vector Machine Degeneracy. In O. Watanabe and T. Yokomori, editors, *Algorithmic Learning Theory*, pages 252–263, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[12] B. Schölkopf and A.J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT press, 2002.

[13] J. R. Zhang, S. Y. Chiu, and L. S. Lan. Non-Uniqueness of Solutions of 1-Norm Support Vector Classification in Dual Form. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 3058–3061, 2008.